IMPLEMENTATION AND EVALUATION OF A DUAL-SENSOR TIME-ADAPTIVE EM ALGORITHM FOR SIGNAL ENHANCEMENT

by

JOHN R. BUCK

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and

WOODS HOLE OCEANOGRAPHIC INSTITUTION

August 1991

Author____
Department of Electrical Engineering, Massachussetts Institute of Technology,
and the Joint Program in Oceanography and Oceanographic Engineering,
Massachusetts Institute of Technology/Woods Hole Oceanographic Institution
August 16, 1991

Certified by____
Alan V. Oppenheim
Distinguished Professor of Electrical Engineering, MIT, Thesis Supervisor

Certified by____
Ehud Weinstein
Adjunct Scientist, WHOI, Thesis Supervisor

Accepted by____
W. Kendall Melville
Chairman, MIT/WHOI Joint Committee on Oceanographic Engineering

Accepted by____
Campbell L. Searle
Chairman, MIT EECS Departmental Committee on Graduate Students

Implementation and Evaluation of a Dual-Sensor Time-Adaptive EM Algorithm for Signal Enhancement

by

John R. Buck

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
and the
WOODS HOLE OCEANOGRAPHIC INSTITUTION
August 1991

Abstract

This thesis describes the implementation and evaluation of an adaptive time-domain algorithm for signal enhancement from multiple-sensor observations. The algorithm is first derived as a noncausal time-domain algorithm, then converted into a causal, recursive form. A more computationally efficient gradient-based parameter estimation step is also presented. The results of several experiments using synthetic data are shown. These experiments first illustrate that the algorithm works on data meeting all the assumptions made by the algorithm, then provide a basis for comparing the performance of the algorithm against the performance of a noncausal frequency-domain algorithm solving the same problem. Finally, an evaluation is made of the performance of the simpler gradient-based parameter estimation step.

Thesis Supervisor: Alan V. Oppenheim
Title: Distinguished Professor of Electrical Engineering, MIT

Thesis Supervisor: Ehud Weinstein
Title: Adjunct Scientist, WHOI

Acknowledgments

Ultimately, my parents made this work possible. Without their love, patience, and understanding, I would not be where I am to write this.

I owe a tremendous amount to Al Oppenheim. Al's insistence on quality and considered work has pushed me to improve beyond what I previously thought were my limits. He has always been able to pat me on the back when I was down, or fire me up when I needed motivating, and as only a good friend can, know which I needed at any given time.

I am also deeply indebted to Udi Weinstein. Udi's combination of patience and contagious enthusiasm was crucial in rekindling my interest in the research at a time when it was lagging. He has been an inspiring example of a conscientious and dedicated scientist for me.

Meir Feder offered his insight and advice on the work here. Meir gave me invaluable help in resurrecting his old programs to allow comparison with the new algorithms discussed in this thesis. He also generously allowed me to tap his expertise and intuitions accumulated during his extensive experience with the EM algorithm.

I am grateful to the Woods Hole Oceanographic Institution for the opportunity to perform most of the experiments described in this thesis while spending a summer there. I will always remember my summer in Woods Hole as one of the most relaxing and productive summers I've ever had. I will also remember it as a place where the wind is often strong.

The Office of Naval Research and the Advanced Research Projects Agency administered by the Office of Naval Research both generously supported me in doing some of the work described in this thesis.

To whatever extent this document is written comprehensibly, it is the responsibility of Fr. William J. O'Malley, S.J. and Fr. Lawrence Wroblewski, S.J., men who patiently attempted to teach me to read comprehendingly, think clearly, and write intelligibly. Professor Theo Theoharis encouraged me in developing my writing style. In addition, Mr. Frank Noce and Mr. Thomas Schaeffer instilled in me an appreciation for the beauty of mathematics and the excitement of science, respectively.

I have been very fortunate to work in the Digital Signal Processing Group at MIT. I have learned from all of my fellow group members, Paul Beckman, Dan Cobra, Steve Isabelle, Jim Njeru, Mike Richard, Stephen Scherock and Kambiz Zangi. However, I have gained the most in conversation and in teaching with Andy Singer, Jim Preisig and Greg Wornell. Over the last two years, they have generously given me advice, assistance, inspiration, and laughter. I have been truly blessed to have companions such as these.

I also wish to thank Deborah Gage, Giovanni Alberti and Giampiero Sciutto, who have all played Virgil to my Dante at many points in my wanderings through the infernal bureaucratic and technical labyrinth of MIT.

A number of people were not directly responsible for the technical work here, but helped keep me sane while doing it. Most prominent among them are my housemates, Brecht, Doug, Hilary, Heather, Kirk, Kevin and Mike. Several other people, Dan, Linda, Mel, Suniti and Theo, were also willing ears and arms when I needed them. These people are all indelibly intertwined in my memories of graduate school and this work, and without their understanding and care, I am skeptical I could have ever finished this.

Finally, and most importantly, as always, *Ad Majorem Dei Gloriam.*

Contents

List of Figures

Chapter 1

Introduction

The problem of signal enhancement appears in a wide range of fields such as seismic exploration, underwater communication, and speech processing. Often, signal enhancement algorithms must track nonstationary signals or environments adaptively. To improve enhancement, many of these algorithms use multiple sensors for their measurements. In most of these algorithms, one sensor primarily measures the desired signal, with some noise corruption. The other sensor measures the noise, possibly coupled with some of the desired signal. These are referred to as the primary and secondary sensors, respectively.

The Least Mean Squares (LMS) algorithm, briefly mentioned in [Widrow 75] and described in detail in [Widrow 85], is one of the better-known two-sensor signal enhancement algorithms. This algorithm possesses the desirable feature that it can be run causally, i.e., the desired signal at some time n_0 can be estimated using only the data up to time n_0. If sufficient computational power is available, the algorithm can run in real time. The LMS algorithm uses the asymmetric system model shown in Figure 1-1. As can be seen in Figure 1-1, the LMS algorithm assumes the desired signal is not coupled to the secondary sensor. The primary sensor measures the desired signal plus the output of some linear filter A applied to the noise. The algorithm assumes the noise is white, Gaussian, and independent of the desired signal, and that the linear filter A is a finite impulse response (FIR) filter. The equations for estimating $s[n]$ and the impulse response of the filter A can be written in a recursive form, so they are easily updated as each new measurement is received from the sensors. When applied to real data, the LMS algorithm generally performs well, but sometimes introduces a reverberation distortion. The presence of the desired signal $s[n]$ in

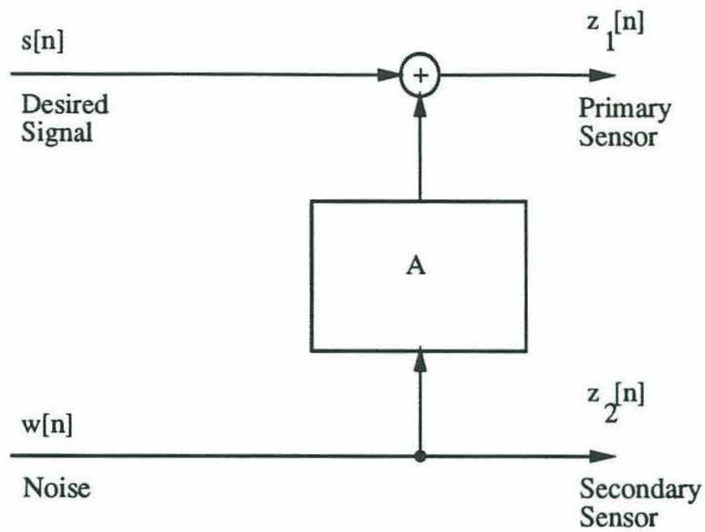

Figure 1-1: Asymmetric System Model Used by Widrow and Stearns

the secondary sensor $z_2[n]$ may cause this distortion.

Algorithms based on more extensive models exist. Feder *et al.* proposed a two-sensor signal-enhancement algorithm based on the Estimate-Maximize (EM) algorithm in [TR 532] and [Feder 89]. This algorithm used the symmetric system model shown in Figure 1-2. This model modifies the model shown in Figure 1-1 by adding another linear filter coupling the desired signal into the secondary sensor. These two linear filters A and B, or coupling filters as they will be referred to in this thesis, are assumed to be FIR filters, with orders q and r, respectively. This means they can be uniquely specified by their impulse responses $a[0], \ldots, a[q]$ and $b[0], \ldots, b[r]$. This model also introduces two additional noise sources, $e_1[n]$ and $e_2[n]$. These low-level, white, Gaussian signals are required by the algorithm for numerical reasons, but also have an intuitively appealing interpretation as sensor noise. The variances of $e_1[n]$ and $e_2[n]$ are g_1 and g_2, respectively, and much smaller than the variance of $w[n]$, g_w. The algorithm presented in [TR 532] assumes that the desired signal $s[n]$ can be modeled as a p^{th}-order autoregressive signal, i.e., any sample can be represented by the following equation:

$$s[n] = -\sum_{k=1}^{p} \alpha[k] s[n-k] + \sqrt{g_s} \cdot u_s[n], \tag{1.1}$$

where $u_s[n]$ is an independent, unit-variance, Gaussian, white noise process, and $\alpha[1], \ldots, \alpha[p]$ are the autoregressive (AR) parameters of the signal.

This algorithm utilizes the EM algorithm to solve the signal enhancement problem.

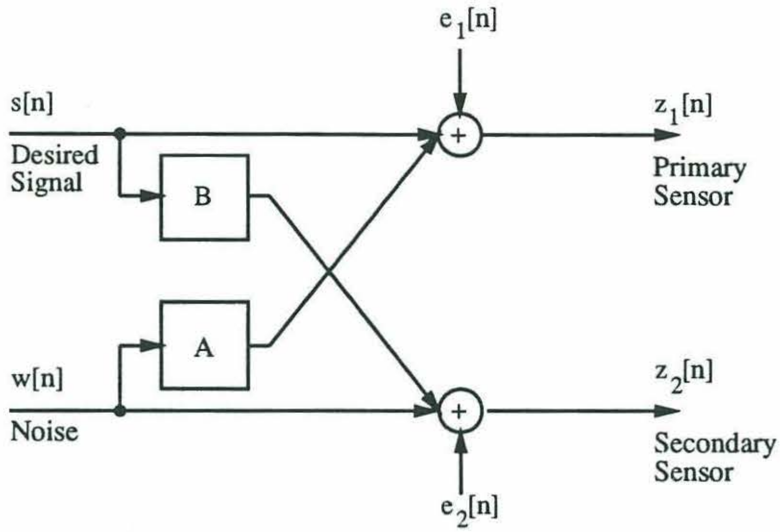

Figure 1-2: Symmetric Model used by Feder *et al.*

The EM algorithm, described in its general form in [Dempster 77], is an iterative parameter estimation algorithm which alternates between two steps. In the Estimate or E-step, the goal is to compute the log-likelihood function of the observed data given the current parameter estimate and observations. In the Maximize or M-step, a new parameter estimate is made which maximizes the log-likelihood function from the E-step. The algorithm then returns to the E-step using this new parameter estimate. It can be shown that estimating then maximizing the expectation of the log-likelihood function of some complete but unobservable data is a sufficient condition for maximizing the likelihood of the observed data under appropriate conditions. This strategy will be covered in more detail in Chapter 2 of this thesis. The complete data can be related to the observed or "incomplete" data though a noninvertible transformation. For the algorithm given in [TR 532], the E-step estimates the current signal value as an intermediate step. This estimate is actually the goal of a signal enhancement algorithm, and the parameter estimates are of only secondary interest.

The algorithm given in [TR 532] uses a noncausal, block approach. Successive iterations are made on a block of data using a Wiener smoother to obtain state estimates. An estimated log-likelihood function is calculated from these estimates. Finally the parameter estimates are updated to maximize this estimated log-likelihood function. According to results presented in [TR 532], this performed favorably when compared with the LMS algorithm using data constructed using the system model of Figure 1-2. Unlike the LMS algorithm, the signal enhancement algorithm described in [TR 532] is inherently noncausal due to the use of a Wiener smoother.

In [TR 560], Weinstein *et al.* describe a causal, time-domain EM algorithm for signal-enhancement for single and multi-sensor systems. While still assuming the system model shown in Figure 1-2 and described above, this algorithm replaces the block frequency-domain Wiener smoother with a time-domain Kalman smoother. Then, by changing this Kalman smoother to a Kalman filter, and replacing iteration indices with time indices, the algorithm was converted to a causal, sequential form. However, no convergence proofs exist for the causal form of the algorithm.

The matrices governing the state-space model of the Kalman filter are sparse, and [TR 560] gives a formulation of the Kalman filter exploiting this sparseness to obtain greater computational efficiency. In addition, an alternative M-step is given based on a maximization strategy utilizing the gradient of the expected value of the log-likelihood function of the complete data. This M-step converges to at least a local, if not global, maximum of the expectation of the log-likelihood function.

Chapter 2 of this thesis reviews the derivation of the algorithm given in [TR 560]. Empirical evidence that this algorithm works on simulated data fitting the model is given in Chapter 3. Chapter 4 presents a comparison of the performance of the causal algorithm of [TR 560] with the noncausal algorithm of [TR 532] using speech in simulated room acoustics. Chapter 5 examines the performance of the algorithm using the computationally simpler gradient-based M-step described in [TR 560]. Finally, Chapter 6 reviews the essential results presented in the thesis, and suggests further topics of investigation based on the work described here.

Chapter 2

The EM Algorithm and the Two-Sensor Model

In this chapter we first review the Estimate-Maximize (EM) Algorithm in general, and then apply the algorithm to the two-sensor model illustrated in Figure 1.2. This chapter also presents the derivation of the gradient-based parameter estimation step. The material presented in this chapter is based largely on the derivations appearing in [TR 560].

2.1 The EM Algorithm

The general formulation of the EM algorithm was presented in [Dempster 77]. The EM algorithm iterates between two steps to converge on a maximum likelihood estimate of an unknown parameter vector using the notion of "complete" and "incomplete" data sets. The observations constitute the incomplete data set, which can be obtained from the complete data set with a noninvertible mapping. In the E-step, the log-likelihood function is calculated for the observed incomplete data. Then, in the M-step, an estimate is obtained for the parameter vector $\boldsymbol{\theta}$ by maximizing the log-likelihood function found in the E-step with respect to $\boldsymbol{\theta}$. It can be shown that maximizing the more conveniently calculated expectation of the complete data ensures finding a parameter estimate which is at least a local, if not a global, maximum of the log-likelihood function of the observations. This allows both the E-step and M-step to concern themselves with only the expected value of the log-likelihood function of the complete data.

To be more precise, let $\boldsymbol{z}$ represent the incomplete data, or observations, and let $\boldsymbol{y}$

be the unobservable but complete data. These quantities are related by a non-invertible transformation $\boldsymbol{z} = H(\boldsymbol{y})$. Thus, some region $\Omega_{\boldsymbol{z}}$ of the sample space of $\boldsymbol{Y}$ maps to a given value of $\boldsymbol{z}$. The probability density functions of the complete and incomplete data indexed on the parameter vector $\boldsymbol{\theta}$ are represented by $f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})$ and $f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$, respectively. The index $\boldsymbol{\theta}$ also has an appealing interpretation as parameterizing the transformation $H(\cdot)$. This interpretation will prove useful in applying the EM algorithm to the signal enhancement problem later in this chapter. In the E-step, $\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ is calculated using the current parameter estimate and observations. Then, in the M-step, the maximum likelihood estimate of the parameter vector $\hat{\boldsymbol{\theta}}_{\text{ML}}$ is found using

$$\widehat{\boldsymbol{\theta}}_{\text{ML}} = \arg\max_{\theta\in\Theta} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}). \tag{2.1}$$

To get an expression for $\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ for the E-step, first consider the probability density function (p.d.f.) for $\boldsymbol{y}$. This p.d.f. can be written as

$$f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) = f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}) f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) \tag{2.2}$$

using Bayes' Rule. Taking the logarithm of both sides of Eq. (2.2) gives

$$\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) = \log f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}) + \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}), \tag{2.3}$$

where $\log(\cdot)$ refers to the natural logarithm function, i.e., logarithm base e. Solving Eq. (2.3) for the log of the p.d.f. of the observations $\boldsymbol{z}$ yields

$$\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) - \log f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}). \tag{2.4}$$

Since $\boldsymbol{y}$ is unobservable, the actual values of $f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})$ or $f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta})$ are not known. However, an appropriate choice for the complete data allows computation of the expectation of these unknown quantities using some specific value of the parameter vector $\boldsymbol{\theta} = \boldsymbol{\theta}'$. Multiplying Eq. (2.4) by $f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}')$, then integrating over $\boldsymbol{Y}$ gives

$$\begin{aligned}\int \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}')d\boldsymbol{Y} &= \int \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}')d\boldsymbol{Y} \\ &\quad - \int \log f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}) f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta}')d\boldsymbol{Y}. \end{aligned} \tag{2.5}$$

This integral leaves the left-hand side of Eq. (2.5) unaltered. The first term on the right-hand side becomes the expectation at $\boldsymbol{\theta} = \boldsymbol{\theta}'$ of $\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})$ given $\boldsymbol{z}$, which will be written as $E_{\boldsymbol{\theta}'}\{\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})|\boldsymbol{z}\}$. Similarly, the second term on that side becomes $E_{\boldsymbol{\theta}'}\{\log f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta})|\boldsymbol{z}\}$. For notational convenience, let

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}') = E_{\boldsymbol{\theta}'}\{\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})|\boldsymbol{z}\} \tag{2.6}$$

and

$$P(\boldsymbol{\theta},\boldsymbol{\theta}') = E_{\boldsymbol{\theta}'}\{\log f_{\boldsymbol{Y}|\boldsymbol{Z}}(\boldsymbol{y}|\boldsymbol{z};\boldsymbol{\theta})|\boldsymbol{z}\}, \tag{2.7}$$

so Eq. (2.5) can be rewritten as

$$\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = Q(\boldsymbol{\theta},\boldsymbol{\theta}') - P(\boldsymbol{\theta},\boldsymbol{\theta}'). \tag{2.8}$$

The M-step seeks to maximize this quantity by choosing $\boldsymbol{\theta}$ at each iteration such that it increases the $Q(\boldsymbol{\theta},\boldsymbol{\theta}')$ term while guaranteeing the $P(\boldsymbol{\theta},\boldsymbol{\theta}')$ term does not increase. Choosing $\boldsymbol{\theta}$ at each iteration to maximize $Q(\boldsymbol{\theta},\boldsymbol{\theta}')$ will always increase $\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$, since Jensen's inequality guarantees any $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ cannot increase $P(\boldsymbol{\theta},\boldsymbol{\theta}')$ [1] This means the E-step only needs to compute $Q(\boldsymbol{\theta},\boldsymbol{\theta}')$, as the M-step can maximize $\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ without considering the $P(\boldsymbol{\theta},\boldsymbol{\theta}')$ term. A proof that this algorithm converges to the maximum likelihood estimate of $\boldsymbol{\theta}$ for "well-behaved" problems is given in Appendix A of [TR 532].

To summarize, the algorithm alternates between the E-step and M-step. At each iteration, the E-step computes an estimate of $Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})$, where $\widehat{\boldsymbol{\theta}}^{(\ell)}$ is the estimate of the parameter vector at the current iteration. Then, the parameter estimate $\widehat{\boldsymbol{\theta}}^{(\ell+1)}$ is chosen to be the value of $\boldsymbol{\theta}$ maximizing $Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})$.

---

[1] Jensen's inequality states that when $f(x)$ and $g(x)$ are probability density functions

$$\int f(x)\log f(x)dx \geq \int f(x)\log g(x)dx.$$

Letting $f(x) = f_{Y|Z}(y|z;\theta')$ and $g(x) = f_{Y|Z}(y|z;\theta)$ gives

$$E_{\theta'}\left\{\log f_{Y|Z}(y|z;\theta')|z\right\} \geq E_{\theta'}\left\{\log f_{Y|Z}(y|z;\theta)|z\right\},$$

or

$$P(\theta',\theta') \geq P(\theta,\theta').$$

2.2 The EM Algorithm and the Two-Sensor Enhancement Problem

This section applies the EM algorithm described in the preceding section to the signal enhancement problem using the model shown in Figure 1-2 and the assumptions described in the previous chapter. This approach yields a set of equations describing the sequential time-domain EM algorithm for signal-enhancement.

2.2.1 Problem Formulation

The EM Algorithm uses the concept of a set of incomplete observations derived from some complete data set. Based on Figure 1-2, the complete and incomplete data are specified as

$$\boldsymbol{z} = \{z_1[n], z_2[n]; n = 1, \ldots, N\} \tag{2.9}$$

and

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{w} \\ \boldsymbol{z} \end{bmatrix}, \tag{2.10}$$

where $\boldsymbol{s}$ and $\boldsymbol{w}$ are defined as

$$\boldsymbol{s} = \{s[n]; n = -r+1, \ldots, N\} \tag{2.11}$$

$$\boldsymbol{w} = \{w[n]; n = -q+1, \ldots, N\}. \tag{2.12}$$

The parameter vector $\boldsymbol{\theta}$ is defined as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{a} \\ \boldsymbol{b} \\ g_s \\ g_w \\ g_1 \\ g_2 \end{bmatrix}, \tag{2.13}$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha[p] \\ \alpha[p-1] \\ \vdots \\ \alpha[1] \end{bmatrix} \tag{2.14}$$

$$\boldsymbol{a} = \begin{bmatrix} a[q] \\ a[q-1] \\ \vdots \\ a[0] \end{bmatrix} \tag{2.15}$$

and

$$\boldsymbol{b} = \begin{bmatrix} b[r] \\ b[r-1] \\ \vdots \\ b[0] \end{bmatrix}. \tag{2.16}$$

Finally, the transformation from the complete to the incomplete data, i.e., $\boldsymbol{z} = H(\boldsymbol{y})$ is, parameterized by $\boldsymbol{\theta}$. These equations can be written using Figure 1-2:

$$z_1[n] = s[n] + e_1[n] + \sum_{k=0}^{q} a[k]w[n-k] \tag{2.17}$$

$$z_2[n] = w[n] + e_2[n] + \sum_{k=0}^{r} b[k]s[n-k]. \tag{2.18}$$

Eqs. (1.1), (2.17), and (2.18) can be rewritten in vector form using the definitions of $\boldsymbol{\alpha}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ from Eqs. (2.14) through (2.16):

$$z_1[n] = s[n] + e_1[n] + \boldsymbol{a}^\top \boldsymbol{w}_q[n] \tag{2.19}$$

$$z_2[n] = w[n] + e_2[n] + \boldsymbol{b}^\top \boldsymbol{s}_r[n] \tag{2.20}$$

$$s[n] = -\boldsymbol{\alpha}^\top \boldsymbol{s}_{p-1}[n-1] + \sqrt{g_s} \cdot u_s[n], \tag{2.21}$$

where

$$\boldsymbol{s}_r[n] = \begin{bmatrix} s[n-r] \\ s[n-r+1] \\ \vdots \\ s[n] \end{bmatrix} \tag{2.22}$$

and

$$\boldsymbol{w}_q[n] = \begin{bmatrix} w[n-q] \\ w[n-q+1] \\ \vdots \\ w[n] \end{bmatrix}. \tag{2.23}$$

2.2.2 The E-step

The goal of the E-step is to calculate an expression for $Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(\ell)})$. From the choice of complete and incomplete data in the previous section, it is straightforward to derive an expression for the expectation of the log-likelihood function of the complete data. To begin, the p.d.f. of $\boldsymbol{y}$ can rewritten using conditioning as

$$f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) = f_{\boldsymbol{S}}(\boldsymbol{s};\boldsymbol{\theta}) \cdot f_{\boldsymbol{W}}(\boldsymbol{w};\boldsymbol{\theta}) \cdot f_{\boldsymbol{Z}|\boldsymbol{S},\boldsymbol{W}}(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{w};\boldsymbol{\theta}) \tag{2.24}$$

since $s[n]$ and $w[n]$ are assumed to be statistically independent. Taking the logarithm of both sides of Eq. (2.24) yields

$$\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) = \log f_{\boldsymbol{S}}(\boldsymbol{s};\boldsymbol{\theta}) + \log f_{\boldsymbol{W}}(\boldsymbol{w};\boldsymbol{\theta}) + \log f_{\boldsymbol{Z}|\boldsymbol{S},\boldsymbol{W}}(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{w};\boldsymbol{\theta}). \tag{2.25}$$

To estimate $\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})$, more detailed expressions are needed for each term on the right-hand side of Eq. (2.25). Using successive conditioning on the first term gives

$$\begin{aligned} \log f_{\boldsymbol{S}}(\boldsymbol{s};\boldsymbol{\theta}) &= \log f_{\boldsymbol{S}}(\boldsymbol{s}_r[0];\boldsymbol{\theta}) + \log f_{\boldsymbol{S}}(s[1]|\boldsymbol{s}_r[0];\boldsymbol{\theta}) + \cdots + \log f_{\boldsymbol{S}}(s[N]|s[N-1];\boldsymbol{\theta}) \\ &= \log f_{\boldsymbol{S}}(\boldsymbol{s}_r[0];\boldsymbol{\theta}) - \frac{N}{2}\log 2\pi g_s - \frac{1}{2g_s}\sum_{k=1}^{N}(s[k] + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{s}_{p-1}[k-1])^2. \end{aligned} \tag{2.26}$$

Using the same strategy on the other terms gives the following expressions for $\log f_{\boldsymbol{W}}(\boldsymbol{w};\boldsymbol{\theta})$ and $\log f_{\boldsymbol{Z}|\boldsymbol{S},\boldsymbol{W}}(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{w};\boldsymbol{\theta})$:

$$\log f_{\boldsymbol{W}}(\boldsymbol{w};\boldsymbol{\theta}) = \log f_{\boldsymbol{W}}(\boldsymbol{w}_q[0];\boldsymbol{\theta}) - \frac{N}{2}\log 2\pi g_w - \frac{1}{2g_w}\sum_{k=1}^{N} w^2[k] \tag{2.27}$$

$$\log f_{\boldsymbol{Z}|\boldsymbol{S},\boldsymbol{W}}(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{w};\boldsymbol{\theta}) = -\frac{N}{2}\log 2\pi g_1 - \frac{1}{2g_1}\sum_{k=1}^{N} e_1^2[k] - \frac{N}{2}\log 2\pi g_2 - \frac{1}{2g_2}\sum_{k=1}^{N} e_2^2[k]. \tag{2.28}$$

The vector equations for the sensors from Eqs. (2.19) and (2.20) allow $e_1[n]$ and $e_2[n]$ to be rewritten as

$$\begin{aligned} e_1[n] &= z_1[n] - s[n] - \boldsymbol{a}^{\mathsf{T}}\boldsymbol{w}_q[n] \qquad (2.29) \\ e_2[n] &= z_2[n] - w[n] - \boldsymbol{b}^{\mathsf{T}}\boldsymbol{s}_r[n]. \qquad (2.30) \end{aligned}$$

Substituting these expressions into Eq. (2.28) gives

$$\begin{aligned}\log f_{\boldsymbol{Z}|\boldsymbol{S},\boldsymbol{W}}(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{w};\boldsymbol{\theta}) &= -\frac{N}{2}\log 2\pi g_1 - \frac{1}{2g_1}\sum_{k=1}^{N}(z_1[k]-s[k]-\boldsymbol{a}^\top \boldsymbol{w}_q[k])^2 \\ &\quad -\frac{N}{2}\log 2\pi g_2 - \frac{1}{2g_2}\sum_{k=1}^{N}(z_2[k]-w[k]-\boldsymbol{b}^\top \boldsymbol{s}_r[k])^2. \qquad (2.31)\end{aligned}$$

Note that several of the terms in Eqs. (2.26), (2.27), and (2.31) are independent of the choice of parameter values, i.e., the $\log 2\pi$ term from several of the equations can combined into one constant. Assuming that the number of observations far exceeds the filter orders for A and B, i.e., $N \gg q$ and r, the initial condition terms in Eqs. (2.26) and (2.27) can be ignored. Making these assumptions, then substituting Eqs. (2.26), (2.27), and (2.31) into Eq. (2.25) yields

$$\begin{aligned}\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) &= C - \frac{N}{2}\log g_s - \frac{1}{2g_s}\sum_{k=1}^{N}(s[k]+\boldsymbol{\alpha}^\top \boldsymbol{s}_{p-1}[k-1])^2 \\ &\quad -\frac{N}{2}\log g_w - \frac{1}{2g_w}\sum_{k=1}^{N} w^2[k] \\ &\quad -\frac{N}{2}\log g_1 - \frac{1}{2g_1}\sum_{k=1}^{N}(z_1[k]-s[k]-\boldsymbol{a}^\top \boldsymbol{w}_q[k])^2 \\ &\quad -\frac{N}{2}\log g_2 - \frac{1}{2g_2}\sum_{k=1}^{N}(z_2[k]-w[k]-\boldsymbol{b}^\top \boldsymbol{s}_r[k])^2. \qquad (2.32)\end{aligned}$$

Because the values of $s[n]$ and $w[n]$ are unknown, expectations must be taken on both sides of Eq. (2.32). These expectation are taken using the current estimate of $\boldsymbol{\theta}$ and the observations. This yields the following equation:

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(\ell)}) &= E_{\widehat{\boldsymbol{\theta}}^{(\ell)}}\{\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})|\boldsymbol{z}\} \\
&= C - \frac{N}{2}\log g_s - \frac{1}{2g_s}\sum_{k=1}^{N}(\widehat{s}^2[k]^{(\ell)} + 2\boldsymbol{\alpha}^\top \widehat{\boldsymbol{s}_{p-1}[k-1]\, s[k]}^{(\ell)} \\
&\qquad + \boldsymbol{\alpha}^\top \widehat{\boldsymbol{s}_{p-1}[k-1]\, \boldsymbol{s}_{p-1}^\top[k-1]}^{(\ell)} \boldsymbol{\alpha}) \\
&\quad - \frac{N}{2}\log g_w - \frac{1}{2g_w}\sum_{k=1}^{N}\widehat{w^{2}}^{(\ell)}[k] \\
&\quad - \frac{N}{2}\log g_1 - \frac{1}{2g_1}\sum_{k=1}^{N}(z_1^2[k] - 2\widehat{s}^{(\ell)}[k]z_1[k] + \widehat{s^{2}}^{(\ell)}[k] \\
&\qquad - 2\boldsymbol{a}^\top \widehat{\boldsymbol{w}}_q^{(\ell)}[k]\, z_1[k] + 2\boldsymbol{a}^\top \widehat{\boldsymbol{w}_q[k]s[k]}^{(\ell)} \\
&\qquad + \boldsymbol{a}^\top \widehat{\boldsymbol{w}_q[k]\boldsymbol{w}_q^\top[k]}^{(\ell)} \boldsymbol{a}) \\
&\quad - \frac{N}{2}\log g_2 - \frac{1}{2g_2}\sum_{k=1}^{N}(z_2^2[k] - 2\widehat{w}^{(\ell)}[k]z_2[k] + \widehat{w^{2}}^{(\ell)}[k] \qquad (2.33) \\
&\qquad - 2\boldsymbol{b}^\top \widehat{\boldsymbol{s}}_r^{(\ell)}[k]z_2[k] + 2\boldsymbol{b}^\top \widehat{\boldsymbol{s}_r[k]w[k]}^{(\ell)} \\
&\qquad + \boldsymbol{b}^\top \widehat{\boldsymbol{s}_r[k]\boldsymbol{s}_r^\top[k]}^{(\ell)} \boldsymbol{b}),
\end{aligned}
$$

where the notation $\widehat{(\cdot)}^{(\ell)}$ denotes $E_{\widehat{\boldsymbol{\theta}}^{(\ell)}}\{\cdot|\boldsymbol{z}\}$, and C is the combination of the terms independent of $\boldsymbol{\theta}$. Besides the elements of $\boldsymbol{\theta}$, this equation only requires estimates of the signal and corrupting noise, plus outer products of these quantities. So, $Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(\ell)})$ can be estimated from the state vector $\boldsymbol{x}[n]$, defined as

$$\boldsymbol{x}[n] = \left[\begin{array}{c} \boldsymbol{s}_r[n] \\ \hline \boldsymbol{w}_q[n] \end{array}\right] = \begin{bmatrix} s[n-r] \\ s[n-r+1] \\ \vdots \\ s[n] \\ w[n-q] \\ w[n-q+1] \\ \vdots \\ w[n] \end{bmatrix}. \tag{2.34}$$

Ultimately, the algorithm needs to be converted to a causal, time-domain, adaptive form. For now, it will be useful to find the estimates of the state vector using a noncausal Kalman smoother. The process of converting the algorithm into a causal, sequential form will be discussed in a later section of this chapter.

2.2.3 State-Space Formulation and Kalman Smoother Equations

This section describes the formulation of the state-space propagation equations as presented in [Gelb 74] and [Anderson 79] for the state vector $\boldsymbol{x}[n]$ and the Kalman smoother equations for estimating $\boldsymbol{x}[n]$. Eqs. (2.19), (2.20), and (2.21) can be written as state-space equations:

$$\boldsymbol{x}[n] = \Phi \boldsymbol{x}[n-1] + G\boldsymbol{u}[n] \tag{2.35}$$

$$\boldsymbol{z}[n] = H\boldsymbol{x}[n] + \boldsymbol{e}[n], \tag{2.36}$$

where the state vector $\boldsymbol{x}[n]$ is defined in Eq. (2.34), and

$$\boldsymbol{z}[n] = \begin{bmatrix} z_1[n] \\ z_2[n] \end{bmatrix} \tag{2.37}$$

$$
\boldsymbol{u}[n] = \begin{bmatrix} u_s[n] \\ u_w[n] \end{bmatrix} \tag{2.38}
$$

$$
\boldsymbol{e}[n] = \begin{bmatrix} e_1[n] \\ e_2[n] \end{bmatrix} \tag{2.39}
$$

$$
G^{\top} = \begin{array}{c}
\begin{array}{cccccccc} & & & (r+1) & & & & \\ & & & \downarrow & & & & \end{array} \\
\begin{bmatrix}
0 & \cdots & 0 & \sqrt{g_s} & 0 & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & \sqrt{g_w}
\end{bmatrix} \\
\begin{array}{cccccccc} & & & & & & & \uparrow \\ & & & & & & & (r+q+2) \end{array}
\end{array} \tag{2.40}
$$

$$
H = \begin{bmatrix}
0 & \cdots & \cdots & 0 & 1 & a[q] & a[q-1] & \cdots & \cdots & a[0] \\
b[r] & b[r-1] & \cdots & \cdots & b[0] & 0 & \cdots & \cdots & 0 & 1
\end{bmatrix} \tag{2.41}
$$

and

$$
\Phi = \left[\begin{array}{c|c} \Phi_s & 0 \\ \hline 0 & \Phi_w \end{array}\right], \tag{2.42}
$$

where

$$
\Phi_s = \begin{bmatrix}
0 & 1 & 0 & \cdots & & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & & & & \vdots \\
\vdots & & \ddots & \ddots & & \ddots & & \vdots \\
\vdots & & & \ddots & & \ddots & \ddots & \vdots \\
\vdots & & & & & \ddots & \ddots & 0 \\
\vdots & & & & & & 0 & 1 \\
\vdots & \cdots & 0 & -\alpha[p] & & -\alpha[p-1] & \cdots & -\alpha[1]
\end{bmatrix} \tag{2.43}
$$

and

$$\Phi_w = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & 0 \\ \vdots & & & & & 0 & 1 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}. \tag{2.44}$$

Also, R is the 2×2 covariance matrix of $\boldsymbol{e}[n]$:

$$R = \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix}. \tag{2.45}$$

In addition, the expectations of the state and its covariance are represented as

$$\boldsymbol{\mu}_{n|k}^{(\ell)} = E_{\hat{\boldsymbol{\theta}}^{(\ell)}}\{\boldsymbol{x}[n]|\boldsymbol{z}[1], \boldsymbol{z}[2], \ldots, \boldsymbol{z}[k]\} \tag{2.46}$$

$$\begin{aligned} P_{n|k}^{(\ell)} &= E_{\hat{\boldsymbol{\theta}}^{(\ell)}}\{[\boldsymbol{x}[n] - \boldsymbol{\mu}_{n|k}^{(\ell)}][\boldsymbol{x}[n] - \boldsymbol{\mu}_{n|k}^{(\ell)}]^\top|\boldsymbol{z}[1], \boldsymbol{z}[2], \ldots, \boldsymbol{z}[k]\} \\ &= E_{\hat{\boldsymbol{\theta}}^{(\ell)}}\{\boldsymbol{x}[n]\boldsymbol{x}^\top[n]|\boldsymbol{z}[1], \boldsymbol{z}[2], \ldots, \boldsymbol{z}[k]\} - \boldsymbol{\mu}_{n|k}^{(\ell)}\boldsymbol{\mu}_{n|k}^{(\ell)\top}. \end{aligned} \tag{2.47}$$

In terms of this notation, the estimates of the state vector and its outer product at the ℓ^{th} iteration are

$$\widehat{\boldsymbol{x}}^{(\ell)}[n] = E_{\hat{\boldsymbol{\theta}}^{(\ell)}}\{\boldsymbol{x}[n]|\boldsymbol{z}\} = \boldsymbol{\mu}_{n|N}^{(\ell)} \tag{2.48}$$

$$\widehat{\boldsymbol{x}[n]\boldsymbol{x}^\top[n]}^{(\ell)} = E_{\hat{\boldsymbol{\theta}}^{(\ell)}}\{\boldsymbol{x}[n]\boldsymbol{x}^\top[n]|\boldsymbol{z}\} = \boldsymbol{\mu}_{n|N}^{(\ell)}\boldsymbol{\mu}_{n|N}^{(\ell)\top} + P_{n|N}^{(\ell)}. \tag{2.49}$$

In writing the Kalman smoother equations, let $(\cdot)^{(\ell)}$ denote the state-space propagation matrix written using $\widehat{\boldsymbol{\theta}}^{(\ell)}$, the estimate of the parameter vector at the ℓ^{th} iteration.

Propagation Equations

For $n = 1, 2, \ldots, N$,

$$\mu^{(\ell)}_{n|n-1} = \Phi^{(\ell)} \mu_{n-1|n-1} \tag{2.50}$$

$$P^{(\ell)}_{n|n-1} = \Phi^{(\ell)} P^{(\ell)}_{n-1|n-1} \Phi^{(\ell)^\top} + G^{(\ell)} G^{(\ell)^\top}. \tag{2.51}$$

Updating Equations

For $n = 1, 2, \ldots, N$,

$$\mu^{(\ell)}_{n|n} = \mu^{(\ell)}_{n|n-1} + K^{(\ell)}_n [z[n] - H^{(\ell)} \mu^{(\ell)}_{n|n-1}] \tag{2.52}$$

$$P^{(\ell)}_{n|n} = [I - K^{(\ell)}_n H^{(\ell)}] \, P^{(\ell)}_{n|n-1}, \tag{2.53}$$

where I is the identity matrix, and $K^{(\ell)}_n$ is the Kalman gain:

$$K^{(\ell)}_n = P^{(\ell)}_{n|n-1} H^{(\ell)^\top} \left[H^{(\ell)} P^{(\ell)}_{n|n-1} H^{(\ell)^\top} + R^{(\ell)} \right]^{-1}. \tag{2.54}$$

Smoothing Equations

For $n = N, N-1, \ldots, 1$,

$$\mu^{(\ell)}_{n-1|N} = \mu^{(\ell)}_{n-1|n-1} + S^{(\ell)}_{n-1} [\mu^{(\ell)}_{n|N} - \Phi^{(\ell)} \mu^{(\ell)}_{n-1|n-1}] \tag{2.55}$$

$$P^{(\ell)}_{n-1|N} = P^{(\ell)}_{n-1|n-1} + S^{(\ell)}_{n-1} [P^{(\ell)}_{n|N} - P^{(\ell)}_{n|n-1}] S^{(\ell)^\top}_{n-1}, \tag{2.56}$$

where

$$S^{(\ell)}_{n-1} \triangleq P^{(\ell)}_{n-1|n-1} \Phi^{(\ell)^\top} P^{(\ell)^{-1}}_{n|n-1}. \tag{2.57}$$

These equations give the time-domain noncausal implementation of the E-step. After also obtaining a noncausal, time-domain formulation for the M-step, both steps will be converted to a causal, sequential form.

2.2.4 The M-step

The M-step solves for the value of the parameter vector which maximizes the estimated log-likelihood function for the complete data calculated in the E-step. As noted earlier, it can be shown that maximizing this function will also maximize the log-likelihood function of the observed incomplete data under the appropriate conditions. Maximizing the value of an expression with respect to a parameter is accomplished by finding the gradient of the expression with regard to the parameter in question, setting this new expression equal to zero, then solving the resulting equation for the parameter in question. This is especially simple in Eq. (2.33) since any element of $\boldsymbol{\theta}$ appears in only one line of the equation, so maximizing $Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(\ell)})$ with respect to any specific parameter reduces to maximizing one line with respect to that parameter. Solving first for $\boldsymbol{\alpha}$, $\boldsymbol{a}$, and $\boldsymbol{b}$, then substituting these expressions into the solutions for g_w, g_1, and g_2 yields nice expressions for the estimates of the variances. Using this strategy to solve for the estimates of the parameters at the $(\ell+1)^{\text{st}}$ iteration gives

$$\widehat{\boldsymbol{\alpha}}^{(\ell+1)} = -\left[\sum_{k=1}^{N} \widehat{\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^{\mathsf{T}}[k-1]}^{(\ell)}\right]^{-1} \sum_{k=1}^{N} \widehat{\boldsymbol{s}_{p-1}[k-1]s[k]}^{(\ell)} \tag{2.58}$$

$$\widehat{\boldsymbol{a}}^{(\ell+1)} = \left[\sum_{k=1}^{N} \widehat{\boldsymbol{w}_q[k]\boldsymbol{w}_q^{\mathsf{T}}[k]}^{(\ell)}\right]^{-1} \sum_{k=1}^{N} \left[\widehat{\boldsymbol{w}}_q^{(\ell)}[k]z_1[k] - \widehat{\boldsymbol{w}_q[k]s[k]}^{(\ell)}\right] \tag{2.59}$$

$$\widehat{\boldsymbol{b}}^{(\ell+1)} = \left[\sum_{k=1}^{N} \widehat{\boldsymbol{s}_r[k]\boldsymbol{s}_r^{\mathsf{T}}[k]}^{(\ell)}\right]^{-1} \sum_{k=1}^{N} \left[\widehat{\boldsymbol{s}}_r^{(\ell)}[k]z_2[k] - \widehat{\boldsymbol{s}_r[k]w[k]}^{(\ell)}\right] \tag{2.60}$$

$$\widehat{g}_s^{(\ell+1)} = \frac{1}{N}\sum_{k=1}^{N}\left(\widehat{s^2}^{(\ell)}[k] + 2\widehat{\boldsymbol{\alpha}}^{(\ell+1)^{\mathsf{T}}} \widehat{\boldsymbol{s}_{p-1}[k-1]s[k]}^{(\ell)} \right. \tag{2.61}$$

$$\left. + \widehat{\boldsymbol{\alpha}}^{(\ell+1)^{\mathsf{T}}} \widehat{\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^{\mathsf{T}}[k-1]}^{(\ell)} \widehat{\boldsymbol{\alpha}}^{(\ell+1)}\right)$$

$$\widehat{g}_1^{(\ell+1)} = \frac{1}{N}\sum_{k=1}^{N}\Bigg(z_1^2[k] - 2\widehat{s}^{(\ell)}[k]z_1[k] + \widehat{s^2}^{(\ell)}[k] \tag{2.62}$$

$$- 2\widehat{\boldsymbol{a}}^{(\ell+1)^\top}\widehat{\boldsymbol{w}}_q^{(\ell)}[k]z_1[k]$$

$$+ 2\widehat{\boldsymbol{a}}^{(\ell+1)^\top}\widehat{\boldsymbol{w}_q[k]s[k]}^{(\ell)}$$

$$+\widehat{\boldsymbol{a}}^{(\ell+1)^\top}\widehat{\boldsymbol{w}_q[k]\boldsymbol{w}_q^\top[k]}^{(\ell)}\widehat{\boldsymbol{a}}^{(\ell+1)}\Bigg)$$

$$\widehat{g}_2^{(\ell+1)} = \frac{1}{N}\sum_{k=1}^{N}\Bigg(z_2^2[k] - 2\widehat{w}^{(\ell)}[k]z_2[k] + \widehat{w^2}^{(\ell)}[k] \tag{2.63}$$

$$- 2\widehat{\boldsymbol{b}}^{(\ell+1)^\top}\widehat{\boldsymbol{s}}_r^{(\ell)}[k]z_2[k]$$

$$+ 2\widehat{\boldsymbol{b}}^{(\ell+1)^\top}\widehat{\boldsymbol{s}_r[k]w[k]}^{(\ell)}$$

$$+\widehat{\boldsymbol{b}}^{(\ell+1)^\top}\widehat{\boldsymbol{s}_r[k]\boldsymbol{s}_r^\top[k]}^{(\ell)}\widehat{\boldsymbol{b}}^{(\ell+1)}\Bigg).$$

Substituting Eqs. (2.58), (2.59), and (2.60) into Eqs. (2.61), (2.62), and (2.63), respectively, then factoring out the transposed values of the estimates of $\boldsymbol{\alpha}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ gives

$$\widehat{g_s}^{(\ell+1)} = \frac{1}{N}\left(\sum_{k=1}^{N}\widehat{s^2}^{(\ell)}[k] \; + \widehat{\boldsymbol{\alpha}}^{(\ell+1)}\widehat{\boldsymbol{s}_{p-1}[k-1]s[k]}^{(\ell)}\right) \tag{2.64}$$

$$\begin{aligned}\widehat{g_1}^{(\ell+1)} &= \frac{1}{N}\sum_{k=1}^{N}\left(z_1^2[k] - 2\widehat{s}^{(\ell)}[k]z_1[k] + \widehat{s^2}^{(\ell)}[k]\right) \\ &\quad - \widehat{\boldsymbol{a}}^{(\ell+1)^\top}\cdot\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\boldsymbol{w}}_q^{(\ell)}[k]z_1[k] - \widehat{\boldsymbol{w}_q[k]s[k]}^{(\ell)}\right) \qquad (2.65)\end{aligned}$$

$$\begin{aligned}\widehat{g_2}^{(\ell+1)} &= \frac{1}{N}\sum_{k=1}^{N}\left(z_2^2[k] - 2\widehat{w}^{(\ell)}[k]z_2[k] + \widehat{w^2}^{(\ell)}[k]\right) \\ &\quad - \widehat{\boldsymbol{b}}^{(\ell+1)^\top}\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\boldsymbol{s}}_r^{(\ell)}[k]z_2[k] - \widehat{\boldsymbol{s}_r[k]w[k]}^{(\ell)}\right). \qquad (2.66)\end{aligned}$$

Finally, the estimate of the variance of the corrupting noise is

$$\widehat{g}_w{}^{(\ell+1)} = \frac{1}{N}\sum_{k=1}^{N} \widehat{w^{2(\ell)}}[k]. \tag{2.67}$$

2.2.5 Gradient-Based Parameter Estimation

The previous section presented solutions for the exact values of the parameters which maximized the estimated log-likelihood function given in Eq. (2.33). Rather than precisely solving for the exact parameter value that maximizes this function at each iteration, an alternative maximization strategy would be to take a step in the direction of the gradient of the estimated log-likelihood function. In order to find the new estimate of a parameter, the gradient of the estimated log-likelihood function with respect to the parameter of interest must be calculated, then this function is evaluated at the current value of the parameter estimate. The new parameter value is then found by starting at the old estimate and taking a small step in the direction of the evaluated gradient. This algorithm can be expressed by the following equation:

$$\widehat{\theta}_i^{(\ell+1)} = \widehat{\theta}_i^{(\ell)} + \delta_i \cdot \frac{1}{N} \cdot \left(\frac{\partial \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})}{\partial \theta_i}\right)_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(\ell)}}, \tag{2.68}$$

where δ_i is the step size for the i^{th} component of $\boldsymbol{\theta}$ and N is the number of samples in the observation interval. So long as the parameters are either stationary or quasi-stationary, and the step size is sufficiently small, this algorithm will converge to at least a local, if not global, maximum of the estimated log-likelihood surface. Obviously, this only gives parameter estimates as good as the estimate of the log-likelihood function. However, this was also the case in the M-step examined in the previous section, where the equations obtained precisely solved the maximization problem for the estimated log-likelihood surface.

The equations for parameter estimation using the gradient strategy are much simpler than those for the precise solution presented in the previous section. Specifically, the gradient-based method does not require any matrix inverses. This savings can be significant, particularly when the filters used in the model are large. The tradeoff between computation and performance for the two parameter estimation strategies is an important issue to be considered, and will be examined in Chapter 5.

The derivation of the gradient-based algorithm begins by taking the gradient of

$\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$. Using Eq. (2.8), the gradient of $\log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ can be expressed as the difference between the gradient of $Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})$ and that of $P(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})$. Jensen's inequality guarantees that $P(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})$ has its maximum at $\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}^{(\ell)}$, so the gradient of $P(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})$ will be zero when evaluated at $\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}^{(\ell)}$. Eliminating this term leaves

$$[\nabla \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})]_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(\ell)}} = \left[\nabla Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})\right]_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(\ell)}}. \tag{2.69}$$

The definition of $Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})$ in Eq. (2.6) allows this to rewritten as

$$\left[\frac{\partial}{\partial\theta_i} Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})\right]_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(\ell)}} = \frac{\partial}{\partial\theta_i} E_{\hat{\boldsymbol{\theta}}^{(\ell)}}\{\log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})|\boldsymbol{z}\}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(\ell)}}. \tag{2.70}$$

Using the Fisher Identity, Eq. (2.70) can be written in the following more computationally tractable form:

$$\left[\frac{\partial}{\partial\theta_i} Q(\boldsymbol{\theta},\widehat{\boldsymbol{\theta}}^{(\ell)})\right]_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(\ell)}} = E_{\hat{\boldsymbol{\theta}}^{(\ell)}}\left\{\frac{\partial}{\partial\theta_i} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta})|\boldsymbol{z}\right\}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(\ell)}}. \tag{2.71}$$

The gradient of the complete data with respect to the various elements of $\boldsymbol{\theta}$ can be derived from Eq. (2.32):

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\alpha}} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) &= -\frac{1}{g_s}\sum_{k=1}^{N}\left(s[k]\boldsymbol{s}_{p-1}^{\top}[k-1] + \boldsymbol{\alpha}^{\top}\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^{\top}[k-1]\right) && (2.72)\\
\frac{\partial}{\partial g_s} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) &= -\frac{N}{2g_s} + \frac{1}{2g_s^2}\sum_{k=1}^{N}\Big(s^2[k] + 2\boldsymbol{\alpha}^{\top}\boldsymbol{s}_{p-1}[k-1]s[k] \\
&\qquad + \boldsymbol{\alpha}^{\top}\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^{\top}[k-1]\boldsymbol{\alpha}\Big) && (2.73)\\
\frac{\partial}{\partial g_w} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) &= -\frac{N}{2g_w} + \frac{1}{2g_w^2}\sum_{k=1}^{N} w^2[k] && (2.74)\\
\frac{\partial}{\partial\boldsymbol{a}} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) &= \frac{1}{g_1}\sum_{k=1}^{N}\left(z_1[k]\boldsymbol{w}_q^{\top}[k] - s[k]\boldsymbol{w}_q^{\top}[k] - \boldsymbol{a}^{\top}\boldsymbol{w}_q[k]\boldsymbol{w}_q^{\top}[k]\right) && (2.75)\\
\frac{\partial}{\partial g_1} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) &= -\frac{N}{2g_1} + \frac{1}{2g_1^2}\sum_{k=1}^{N}\Big(z_1^2[k] - 2z_1[k]s[k] + s^2[k] - 2\boldsymbol{a}^{\top}\boldsymbol{w}_q[k]z_1[k] \\
&\qquad + 2\boldsymbol{a}^{\top}\boldsymbol{w}_q[k]s[k] + \boldsymbol{a}^{\top}\boldsymbol{w}_q[k]\boldsymbol{w}_q^{\top}[k]\boldsymbol{a}\Big) && (2.76)\\
\frac{\partial}{\partial\boldsymbol{b}} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) &= \frac{1}{g_2}\sum_{k=1}^{N}\left(z_2[k]\boldsymbol{s}_r^{\top}[k] - w[k]\boldsymbol{s}_r[k]^{\top} - \boldsymbol{b}^{\top}\boldsymbol{s}_r[k]\boldsymbol{s}_r^{\top}[k]\right) && (2.77)
\end{aligned}$$

$$\frac{\partial}{\partial g_2} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\boldsymbol{\theta}) = -\frac{N}{2g_2} + \frac{1}{2g_2^2}\sum_{k=1}^{N}\bigg(z_2^2[k] - 2z_2[k]w[k] + w^2[k] - 2\boldsymbol{b}^\top \boldsymbol{s}_r[k]z_2[k] + 2\boldsymbol{b}^\top \boldsymbol{s}_r[k]w[k] + \boldsymbol{b}^\top \boldsymbol{s}_r[k]\boldsymbol{s}_r^\top[k]\boldsymbol{b}\bigg). \tag{2.78}$$

The gradient of the incomplete data is obtained by taking the expectation of Eqs. (2.72) through (2.78):

$$\frac{\partial}{\partial \boldsymbol{\alpha}} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = -\frac{1}{g_s}\sum_{k=1}^{N}\left(\widehat{s[k]\boldsymbol{s}_{p-1}^\top[k-1]} + \boldsymbol{\alpha}^\top \widehat{\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^\top[k-1]}\right) \tag{2.79}$$

$$\frac{\partial}{\partial g_s} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = -\frac{N}{2g_s} + \frac{1}{2g_s^2}\sum_{k=1}^{N}\bigg(\widehat{s^2}[k] + 2\boldsymbol{\alpha}^\top \widehat{\boldsymbol{s}_{p-1}[k-1]s[k]} + \boldsymbol{\alpha}^\top \widehat{\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^\top[k-1]}\,\boldsymbol{\alpha}\bigg) \tag{2.80}$$

$$\frac{\partial}{\partial g_w} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = -\frac{N}{2g_w} + \frac{1}{2g_w^2}\sum_{k=1}^{N}\widehat{w^2}[k] \tag{2.81}$$

$$\frac{\partial}{\partial \boldsymbol{a}} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = \frac{1}{g_1}\sum_{k=1}^{N}\left(z_1[k]\widehat{\boldsymbol{w}}_q^\top[k] - \widehat{s[k]\boldsymbol{w}_q^\top[k]} - \boldsymbol{a}^\top \widehat{\boldsymbol{w}_q[k]\boldsymbol{w}_q^\top[k]}\right) \tag{2.82}$$

$$\frac{\partial}{\partial g_1} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = -\frac{N}{2g_1} + \frac{1}{2g_1^2}\sum_{k=1}^{N}\bigg(z_1^2[k] - 2z_1[k]\widehat{s}[k] + \widehat{s^2}[k] - 2\boldsymbol{a}^\top \widehat{\boldsymbol{w}}_q[k]z_1[k] + 2\boldsymbol{a}^\top \widehat{\boldsymbol{w}_q[k]s[k]} + \boldsymbol{a}^\top \widehat{\boldsymbol{w}_q[k]\boldsymbol{w}_q^\top[k]}\,\boldsymbol{a}\bigg) \tag{2.83}$$

$$\frac{\partial}{\partial \boldsymbol{b}} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = \frac{1}{g_2}\sum_{k=1}^{N}\left(z_2[k]\widehat{\boldsymbol{s}}_r^\top[k] - \widehat{w[k]\boldsymbol{s}_r[k]}^\top - \boldsymbol{b}^\top \widehat{\boldsymbol{s}_r[k]\boldsymbol{s}_r^\top[k]}\right) \tag{2.84}$$

$$\frac{\partial}{\partial g_2} \log f_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}) = -\frac{N}{2g_2} + \frac{1}{2g_2^2}\sum_{k=1}^{N}\bigg(z_2^2[k] - 2z_2[k]\widehat{w}[k] + \widehat{w^2}[k] - 2\boldsymbol{b}^\top \widehat{\boldsymbol{s}}_r[k]z_2[k] + 2\boldsymbol{b}^\top \widehat{\boldsymbol{s}_r[k]w[k]} + \boldsymbol{b}^\top \widehat{\boldsymbol{s}_r[k]\boldsymbol{s}_r^\top[k]}\,\boldsymbol{b}\bigg), \tag{2.85}$$

where $\widehat{(\cdot)}$ signifies $E_{\hat{\boldsymbol{\theta}}}\{\cdot|\boldsymbol{z}\}$ as in Eq. (2.33).

As in the precise solution M-step, the only quantities needed to compute the gradients are $\widehat{\boldsymbol{x}}[n]$ and $\widehat{\boldsymbol{x}[n]\boldsymbol{x}^\top[n]}$. Plugging Eqs. (2.79) through (2.85) into Eq. (2.68) gives the gradient-based M-step for parameter estimation:

$$\widehat{\boldsymbol{\alpha}}^{(\ell+1)} = \widehat{\boldsymbol{\alpha}}^{(\ell)} - \frac{\delta_\alpha}{\widehat{g}_s^{(\ell)}} \cdot \frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\boldsymbol{s}_{p-1}[k-1]s[k]}^{(\ell)} + \widehat{\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^{\mathsf{T}}[k-1]}\,\widehat{\boldsymbol{\alpha}}^{(\ell)}\right) \tag{2.86}$$

$$\widehat{g}_s^{(\ell+1)} = \left(1-\frac{\tilde{\delta}_s}{2}\right)\widehat{g}_s^{(\ell)} + \frac{\tilde{\delta}_s}{2}\cdot\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{s^2}^{(\ell)}[k] + 2\widehat{\boldsymbol{\alpha}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{s}_{p-1}[k-1]s[k]}^{(\ell)}\right.$$

$$\left. + \,\widehat{\boldsymbol{\alpha}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{s}_{p-1}[k-1]\boldsymbol{s}_{p-1}^{\mathsf{T}}[k-1]}\,\widehat{\boldsymbol{\alpha}}^{(\ell)}\right) \tag{2.87}$$

$$\widehat{g}_w^{(\ell+1)} = \left(1-\frac{\tilde{\delta}_w}{2}\right)\widehat{g}_w^{(\ell)} + \frac{\tilde{\delta}_w}{2}\cdot\frac{1}{N}\sum_{k=1}^{N}\widehat{w^2}^{(\ell)}[k] \tag{2.88}$$

$$\widehat{\boldsymbol{a}}^{(\ell+1)} = \widehat{\boldsymbol{a}}^{(\ell)} + \frac{\tilde{\delta}_a}{\widehat{g}_1^{(\ell)}}\cdot\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\boldsymbol{w}}_q^{(\ell)}[k]z_1[k] - \widehat{\boldsymbol{w}_q[k]s[k]}^{(\ell)} - \widehat{\boldsymbol{w}_q[k]\boldsymbol{w}_q^{\mathsf{T}}[k]}\,\widehat{\boldsymbol{a}}^{(\ell)}\right) \tag{2.89}$$

$$\widehat{g}_1^{(\ell+1)} = \left(1-\frac{\tilde{\delta}_1}{2}\right)\widehat{g}_1^{(\ell)} + \frac{\tilde{\delta}_1}{2}\cdot\frac{1}{N}\sum_{k=1}^{N}\left(z_1^2[k] - 2z_1[k\widehat{s}^{(\ell)}[k] + \widehat{s^2}^{(\ell)}[k]\right.$$

$$\left. - \,2\widehat{\boldsymbol{a}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{w}}_q^{(\ell)}[k]z_1[k] + 2\widehat{\boldsymbol{a}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{w}_q[k]s[k]}^{(\ell)} + \widehat{\boldsymbol{a}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{w}_q[k]\boldsymbol{w}_q^{\mathsf{T}}[k]}\,\widehat{\boldsymbol{a}}^{(\ell)}\right) \tag{2.90}$$

$$\widehat{\boldsymbol{b}}^{(\ell+1)} = \widehat{\boldsymbol{b}}^{(\ell)} + \frac{\tilde{\delta}_b}{\widehat{g}_2^{(\ell)}}\cdot\frac{1}{N}\sum_{k=1}^{N}\left(\widehat{\boldsymbol{s}}_r^{(\ell)}[k]z_2[k] - \widehat{\boldsymbol{s}_r[k]w[k]}^{(\ell)} - \widehat{\boldsymbol{s}_r[k]\boldsymbol{s}_r^{\mathsf{T}}[k]}^{(\ell)}\widehat{\boldsymbol{b}}\right) \tag{2.91}$$

$$\widehat{g}_2^{(\ell+1)} = \left(1-\frac{\tilde{\delta}_2}{2}\right)\widehat{g}_2^{(\ell)} + \frac{\tilde{\delta}_2}{2}\cdot\frac{1}{N}\sum_{k=1}^{N}\left(z_2^2[k] - 2z_2[k]\widehat{w}^{(\ell)}[k] + \widehat{w^2}^{(\ell)}[k]\right.$$

$$\left. - \,2\widehat{\boldsymbol{b}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{s}}_r^{(\ell)}[k]z_2[k] + 2\widehat{\boldsymbol{b}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{s}_r[k]w[k]}^{(\ell)} + \widehat{\boldsymbol{b}}^{(\ell)\mathsf{T}}\widehat{\boldsymbol{s}_r[k]\boldsymbol{s}_r^{\mathsf{T}}[k]}\,\widehat{\boldsymbol{b}}^{(\ell)}\right), \tag{2.92}$$

where $\tilde{\delta}_s = \delta_s/(\widehat{g}_s{}^{(\ell)})^2$, and $\tilde{\delta}_w$, $\tilde{\delta}_1$, and $\tilde{\delta}_2$ are all defined similarly. Note that this M-step can be used with exactly the same E-step described earlier, as the same state vector is still the only information needed for parameter estimation.

2.2.6 The Sequential Form of the Algorithm

The algorithm described thus far in this chapter has been an iterative, noncausal, time-domain solution. This already differs from the algorithm given in [TR 532], which was a block solution implemented in the frequency-domain. This section presents a strategy to convert the noncausal, block, time-domain algorithm presented already in this chapter into a casual, sequential, time-domain algorithm.

The Sequential Form of the E-Step

To make the E-step causal and recursive, the Kalman smoother used for state estimation is replaced by a Kalman filter. The equations for the Kalman filter are those given by the Propagation and Updating equations, Eqs. (2.50) through (2.54), but the Smoother equations are omitted.

Propagation Equations

For $n = 1, 2, \ldots, N$,

$$\begin{aligned} \mu^{(\ell)}_{n|n-1} &= \Phi^{(\ell)} \mu_{n-1|n-1} && (2.93) \\ P^{(\ell)}_{n|n-1} &= \Phi^{(\ell)} P^{(\ell)}_{n-1|n-1} \Phi^{(\ell)^{\top}} + G^{(\ell)} G^{(\ell)^{\top}} . && (2.94) \end{aligned}$$

Updating Equations

For $n = 1, 2, \ldots, N$,

$$\begin{aligned} \mu^{(\ell)}_{n|n} &= \mu^{(\ell)}_{n|n-1} + K^{(\ell)}_n [z[n] - H^{(\ell)} \mu^{(\ell)}_{n|n-1}] && (2.95) \\ P^{(\ell)}_{n|n} &= [I - K^{(\ell)}_n H^{(\ell)}] \, P^{(\ell)}_{n|n-1}, && (2.96) \end{aligned}$$

where I is the identity matrix, and $K^{(\ell)}_n$ is the Kalman gain

$$K^{(\ell)}_n = P^{(\ell)}_{n|n-1} H^{(\ell)^{\top}} \left[H^{(\ell)} P^{(\ell)}_{n|n-1} H^{(\ell)^{\top}} + R^{(\ell)} \right]^{-1} . \tag{2.97}$$

The sparse form of the matrices in the Kalman state-space model, i.e., Eqs. (2.35) and (2.36), can be exploited to obtain a more computationally efficient implementation of the Kalman filter. This efficient version is outlined below. Interested readers can find a derivation of these results in Appendix A of [TR 560].

First, $\mu_{n-1|n-1}$ is partitioned as

$$\mu_{n-1|n-1} = \left[\begin{array}{c} \mu_s \\ \hline \mu_w \end{array} \right] \begin{array}{l} \updownarrow r+1 \\ \updownarrow q+1 \end{array} = \left[\begin{array}{c} \mu_1 \\ \hline \mu_2 \\ \hline \mu_3 \\ \hline \mu_4 \end{array} \right] \begin{array}{l} \updownarrow 1 \\ \updownarrow r \\ \updownarrow 1 \\ \updownarrow q \end{array} . \tag{2.98}$$

Then, $\boldsymbol{\mu}_p$ is defined as the lower $p \times 1$ subvector of $\boldsymbol{\mu}_2$:

$$\boldsymbol{\mu}_2 = \left[\begin{array}{c} \\ \hline \boldsymbol{\mu}_p \end{array}\right] \begin{array}{l} \\ \updownarrow p \end{array} . \tag{2.99}$$

Next, $P_{n-1|n-1}$ is partitioned as

$$\begin{aligned} P_{n-1|n-1} &= \begin{array}{c} \left[\begin{array}{c|c} P_{ss} & P_{sw} \\ \hline P_{sw}^{\top} & P_{ww} \end{array}\right] \begin{array}{l} \updownarrow r+1 \\ \updownarrow q+1 \end{array} \\ \begin{array}{cc} \xleftrightarrow[r+1]{} & \xleftrightarrow[q+1]{} \end{array} \end{array} \\ &= \begin{array}{c} \left[\begin{array}{c|c|c|c} P_{11} & P_{12} & P_{13} & P_{14} \\ \hline P_{12}^{\top} & P_{22} & P_{23} & P_{24} \\ \hline P_{13}^{\top} & P_{23}^{\top} & P_{33} & P_{34} \\ \hline P_{14}^{\top} & P_{24}^{\top} & P_{34}^{\top} & P_{44} \end{array}\right] \begin{array}{l} \updownarrow 1 \\ \updownarrow r \\ \updownarrow 1 \\ \updownarrow q \end{array} \\ \begin{array}{cccc} \xleftrightarrow[1]{} & \xleftrightarrow[r]{} & \xleftrightarrow[1]{} & \xleftrightarrow[q]{} \end{array} \end{array} . \end{aligned} \tag{2.100}$$

Several submatrices of $P_{n-1|n-1}$ also need to be partitioned out. Let Γ_p be the right-most p columns of P_{22}:

$$P_{22} = \left[\begin{array}{c|c} \quad\quad & \underset{\xleftrightarrow[p]{}}{\Gamma_p} \end{array}\right] \updownarrow r , \tag{2.101}$$

and Γ_{pp} be the bottom p rows of Γ_p:

$$\Gamma_p = \left[\begin{array}{c} \\ \hline \underset{\xleftrightarrow[p]{}}{\Gamma_{pp}} \end{array}\right] \begin{array}{l} \\ \updownarrow p \end{array} . \tag{2.102}$$

Λ_p is defined as the bottom p rows of P_{24}:

$$P_{24} = \left[\begin{array}{c} \\ \hline \underset{\xleftrightarrow[q]{}}{\Lambda_p} \end{array}\right] \begin{array}{l} \\ \updownarrow p \end{array} . \tag{2.103}$$

Finally, the filter impulse responses are partitioned as shown below:

$$\boldsymbol{a} = \left[\begin{array}{c} \boldsymbol{a}_1 \\ \hline a_0 \end{array}\right] \begin{array}{l} \updownarrow q \\ \updownarrow 1 \end{array} \tag{2.104}$$

$$\boldsymbol{b} = \left[\begin{array}{c} \boldsymbol{b}_1 \\ \hline b_0 \end{array}\right] \begin{array}{l} \updownarrow r \\ \updownarrow 1 \end{array} . \tag{2.105}$$

In the following equations, $\boldsymbol{\theta}$ will be understood to represent $\widehat{\boldsymbol{\theta}}[n]$ to eliminate notational clutter. Using the partitions given above, Eqs. (2.93) through (2.96) can be rewritten as:

Propagation Equations

For $n = 1, 2, \ldots, N$,

$$\boldsymbol{\mu}_{n|n-1} = \left[\begin{array}{c} \boldsymbol{\mu}_2 \\ \hline -\boldsymbol{\alpha}^\top \boldsymbol{\mu}_p \\ \hline \boldsymbol{\mu}_4 \\ \hline 0 \end{array}\right] \begin{array}{l} \updownarrow r \\ \updownarrow 1 \\ \updownarrow q \\ \updownarrow 1 \end{array} \tag{2.106}$$

$$P_{n|n-1} = \begin{array}{c} \left[\begin{array}{c|c|c|c} P_{22} & -\Gamma_p \boldsymbol{\alpha} & P_{24} & \mathbf{0} \\ \hline -(\Gamma_p \boldsymbol{\alpha})^\top & \boldsymbol{\alpha}^\top \Gamma_{pp} \boldsymbol{\alpha} + g_s & -\boldsymbol{\alpha}^\top \Lambda_p & 0 \\ \hline P_{24}^\top & -\Lambda_p^\top \boldsymbol{\alpha} & P_{44} & \mathbf{0} \\ \hline \mathbf{0} & 0 & \mathbf{0} & g_w \end{array}\right] \begin{array}{l} \updownarrow r \\ \updownarrow 1 \\ \updownarrow q \\ \updownarrow 1 \end{array} \\ \begin{array}{cccc} \xleftrightarrow{\;r\;} & \xleftrightarrow{\;1\;} & \xleftrightarrow{\;q\;} & \xleftrightarrow{\;1\;} \end{array} \end{array} . \tag{2.107}$$

Updating Equations

For $n = 1, 2, \ldots, N$,

$$\boldsymbol{\mu}_{n|n} = \boldsymbol{\mu}_{n|n-1} + K_n \cdot \left[\begin{array}{c} z_1[n] + \boldsymbol{\alpha}^\top \boldsymbol{\mu}_p - \boldsymbol{a}_1^\top \boldsymbol{\mu}_4 \\ \hline z_2[n] + b_0 \boldsymbol{\alpha}^\top \boldsymbol{\mu}_p - \boldsymbol{b}_1^\top \boldsymbol{\mu}_2 \end{array}\right] \tag{2.108}$$

$$P_{n|n} = P_{n|n-1} - K_n D_n^\top . \tag{2.109}$$

The Kalman gain, K_n, is defined as

$$K_n = D_n F_n^{-1}, \tag{2.110}$$

where

$$D_n = \begin{array}{cc|cc} & \left[\begin{array}{c|c} -\Gamma_p\boldsymbol{\alpha} + P_{24}\boldsymbol{a}_1 & P_{22}\boldsymbol{b}_1 - \Gamma_p\boldsymbol{\alpha}\cdot b_0 \\ \hline \boldsymbol{\alpha}^\top\Gamma_{pp}\boldsymbol{\alpha} + g_s - \boldsymbol{\alpha}^\top\Lambda_p\boldsymbol{a}_1 & -\boldsymbol{\alpha}^\top\Gamma_p^\top\boldsymbol{b}_1 + b_0(\boldsymbol{\alpha}^\top\Gamma_{pp}\boldsymbol{\alpha} + g_s) \\ \hline -\Lambda_p^\top\boldsymbol{\alpha} + P_{44}\boldsymbol{a}_1 & P_{24}^\top\boldsymbol{b}_1 - \Lambda_p^\top\boldsymbol{\alpha}\cdot b_0 \\ \hline a_0\cdot g_w & g_w \end{array}\right] & \begin{array}{c} \updownarrow r \\ \updownarrow 1 \\ \updownarrow q \\ \updownarrow 1 \end{array} \end{array} \tag{2.111}$$

$$\begin{array}{cc} \xleftrightarrow{\quad 1 \quad} & \xleftrightarrow{\quad 1 \quad} \end{array}$$

and F_n is the 2×2 symmetric matrix

$$F_n = \begin{bmatrix} f_{11} & f_{12} \\ f_{12} & f_{22} \end{bmatrix}, \tag{2.112}$$

where

$$f_{11} = \boldsymbol{a}_1^\top P_{44}\boldsymbol{a}_1 - 2\boldsymbol{a}_1^\top\Lambda_p^\top\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top\Gamma_{pp}\boldsymbol{\alpha} + g_s + a_0^2 g_w + g_1 \tag{2.113}$$

$$f_{22} = \boldsymbol{b}_1^\top P_{22}\boldsymbol{b}_1 - 2b_0\cdot\boldsymbol{b}_1^\top\Gamma_p\boldsymbol{\alpha} + b_0^2(\boldsymbol{\alpha}^\top\Gamma_{pp}\boldsymbol{\alpha} + g_s) + g_w + g_2 \tag{2.114}$$

$$f_{12} = \boldsymbol{a}_1^\top P_{24}^\top\boldsymbol{b}_1 - b_0\cdot\boldsymbol{a}_1^\top\Lambda_p^\top\boldsymbol{\alpha} - \boldsymbol{b}_1^\top\Gamma_p\boldsymbol{\alpha} + b_0(\boldsymbol{\alpha}^\top\Gamma_{pp}\boldsymbol{\alpha} + g_s) + a_0 g_w. \tag{2.115}$$

The Sequential Form of the Exact Solution M-step

The strategy for converting the M-step to a casual, sequential form is to replace iteration indices with time indices. Specifically, the estimates of quantities at the ℓ^{th} iteration are replaced with estimates of the same quantity given the observations up to the current time n. Thus, all expressions of the form $\widehat{(\cdot)}^{(\ell)}$ become $\widehat{(\cdot)}|\boldsymbol{z}[1], \boldsymbol{z}[2], \ldots, \boldsymbol{z}[n]$. In addition, the summations over the entire data set are restricted to be causal summations up to the current time, and a geometric weighting factor is also introduced into the sums. Setting this factor to be close to, but less than, one, should improve the adaptive characteristic of the algorithm. This discounts older data and weights recent data more highly, so the algorithm should track changes in the data more quickly than if all data was equally weighted. While these changes are all intuitively reasonable, they cannot be justified theoretically. No proof of convergence exists for the causal, sequential form of the algorithm. Chapter 3 of this thesis presents an experiment that provides empirical evidence that this strategy yields a useful causal algorithm for signal enhancement.

In the equations describing the sequential form of the exact solution M-step below, the notation $\widehat{(\cdot)}[k|\ell]$ will denote the estimate of the quantity at time k given the observations up to time ℓ:

$$\widehat{\boldsymbol{\alpha}}[n+1] = -\left[\sum_{k=1}^{n} \gamma_s^{n-k} \widehat{\boldsymbol{s}_{p-1}[k-1|k]\boldsymbol{s}_{p-1}^{\mathsf{T}}[k-1|k]}\right]^{-1} \cdot \left[\sum_{k=1}^{n} \gamma_s^{n-k} \widehat{\boldsymbol{s}_{p-1}[k-1|k]s[k|k]}\right] \tag{2.116}$$

$$\widehat{g}_s[n+1] = \frac{1}{\sum_{k=1}^{n} \gamma_s^{n-k}} \left(\sum_{k=1}^{n} \gamma_s^{n-k} \widehat{s^2}[k|k] + \widehat{\boldsymbol{\alpha}}^{\mathsf{T}}[n+1] \sum_{k=1}^{n} \gamma_s^{n-k} \widehat{\boldsymbol{s}_{p-1}[k-1|k]s[k|k]} \right) \tag{2.117}$$

$$\widehat{g}_w[n+1] = \frac{1}{\sum_{k=1}^{n} \gamma_w^{n-k}} \cdot \sum_{k=1}^{n} \gamma_w^{n-k} \widehat{w^2}[k|k] \tag{2.118}$$

$$\widehat{\boldsymbol{a}}[n+1] = \left[\sum_{k=1}^{n} \gamma_a^{n-k} \widehat{\boldsymbol{w}_q[k|k]\boldsymbol{w}_q^{\mathsf{T}}[k|k]}\right]^{-1} \sum_{k=1}^{n} \gamma_a^{n-k} \left[\widehat{\boldsymbol{w}}_q[k|k]z_1[k] - \widehat{\boldsymbol{w}_q[k|k]s[k|k]}\right] \tag{2.119}$$

$$\widehat{g}_1[n+1] = \frac{1}{\sum_{k=1}^{n} \gamma_a^{n-k}} \left(\sum_{k=1}^{n} \gamma_a^{n-k} \left[z_1^2[k] - 2\widehat{s}[k|k]z_1[k] + \widehat{s^2}[k|k] \right] - \widehat{\boldsymbol{a}}^{\mathsf{T}}[n+1] \sum_{k=1}^{n} \gamma_a^{n-k} \left[\widehat{\boldsymbol{w}}_q[k|k]z_1[k] - \widehat{\boldsymbol{w}_q[k|k]s[k|k]} \right] \right) \tag{2.120}$$

$$\widehat{\boldsymbol{b}}[n+1] = \left[\sum_{k=1}^{n} \gamma_b^{n-k} \widehat{\boldsymbol{s}_r[k|k]\boldsymbol{s}_r^{\mathsf{T}}[k|k]}\right]^{-1} \sum_{k=1}^{n} \gamma_b^{n-k} \left[\widehat{\boldsymbol{s}_r}[k|k]z_2[k] - \widehat{\boldsymbol{s}_r[k|k]w[k|k]}\right] \tag{2.121}$$

$$\widehat{g}_2[n+1] = \frac{1}{\sum_{k=1}^{n} \gamma_b^{n-k}} \left(\sum_{k=1}^{n} \gamma_b^{n-k} [z_2^2[k] - 2\widehat{w}[k|k] z_2[k] + \widehat{w^2}[k|k]] \right.$$
$$\left. - \widehat{\boldsymbol{b}}^{\mathsf{T}}[n+1] \sum_{k=1}^{n} \gamma_b^{n-k} \left[\widehat{\boldsymbol{s}}_r[k|k] z_2[k] - \widehat{\boldsymbol{s}_r[k|k] w[k|k]} \right] \right). \quad (2.122)$$

These equations can be expressed in terms of the following recursively computable quantities:

$$R_{11}[n] = \sum_{k=1}^{n} \gamma_s^{n-k} \widehat{\boldsymbol{s}_{p-1}[k-1|k] \boldsymbol{s}_{p-1}^{\mathsf{T}}[k-1|k]}$$
$$= \gamma_s R_{11}[n-1] + \widehat{\boldsymbol{s}_{p-1}[n-1|n] \boldsymbol{s}_{p-1}^{\mathsf{T}}[n-1|n]} \quad (2.123)$$
$$R_{12}[n] = \sum_{k=1}^{n} \gamma_s^{n-k} \widehat{\boldsymbol{s}_{p-1}[k-1|k] s[k|k]}$$
$$= \gamma_s R_{12}[n-1] + \widehat{\boldsymbol{s}_{p-1}[n-1|n] s[n|n]} \quad (2.124)$$
$$R_{22}[n] = \sum_{k=1}^{n} \gamma_s^{n-k} \widehat{s^2}[k|k] = \gamma_s R_{22}[n-1] + \widehat{s^2}[n|n] \quad (2.125)$$
$$Q_{11}[n] = \sum_{k=1}^{n} \gamma_w^{n-k} \widehat{w^2}[k|k] = \gamma_w Q_{11}[n-1] + \widehat{w^2}[n|n] \quad (2.126)$$
$$A_{11}[n] = \sum_{k=1}^{n} \gamma_a^{n-k} \widehat{\boldsymbol{w}_q[k|k] \boldsymbol{w}_q^{\mathsf{T}}[k|k]}$$
$$= \gamma_a A_{11}[n-1] + \widehat{\boldsymbol{w}_q[n|n] \boldsymbol{w}_q^{\mathsf{T}}[n|n]} \quad (2.127)$$
$$A_{12}[n] = \sum_{k=1}^{n} \gamma_a^{n-k} \left(\widehat{\boldsymbol{w}}_q[k|k] z_1[k] - \widehat{\boldsymbol{w}_q[k|k] s[k|k]} \right)$$
$$= \gamma_a A_{12}[n-1] + \widehat{\boldsymbol{w}}_q[n|n] z_1[n] - \widehat{\boldsymbol{w}_q[n|n] s[n|n]} \quad (2.128)$$
$$A_{22}[n] = \sum_{k=1}^{n} \gamma_a^{n-k} (z_1^2[k] - 2\widehat{s}[k|k] z_1[k] + \widehat{s^2}[k|k])$$
$$= \gamma_a A_{22}[n-1] + z_1^2[n] - 2\widehat{s}[n|n] z_1[n] + \widehat{s^2}[n|n] \quad (2.129)$$
$$B_{11}[n] = \sum_{k=1}^{n} \gamma_b^{n-k} \widehat{\boldsymbol{s}_r[k|k] \boldsymbol{s}_r^{\mathsf{T}}[k|k]} = \gamma_b B_{11}[n-1] + \widehat{\boldsymbol{s}_r[n|n] \boldsymbol{s}_r^{\mathsf{T}}[n|n]} \quad (2.130)$$

$$B_{12}[n] = \sum_{k=1}^{n} \gamma_b^{n-k}\left(\widehat{s}_r[k|k]z_2[k] - \widehat{s_r[k]w[k|k]}\right)$$

$$= \gamma_b B_{12}[n-1] + \widehat{s}_r[n|n]z_2[n] - \widehat{s_r[n|n]w[n|n]} \tag{2.131}$$

$$B_{22}[n] = \sum_{k=1}^{n} \gamma_b^{n-k}(z_2^2[k] - 2\widehat{w}[k|k]z_2[k] + \widehat{w^2}[k|k])$$

$$= \gamma_b B_{22}[n-1] + z_2^2[n] - 2\widehat{w}[n|n]z_2[n] + \widehat{w^2}[n|n]. \tag{2.132}$$

Using the quantities defined in Eqs. (2.123) through (2.132) to rewrite Eqs. (2.116) through (2.122) gives

$$\widehat{\boldsymbol{\alpha}}[n+1] = -R_{11}^{-1}[n]R_{12}[n]$$

$$\widehat{g}_s[n+1] = \frac{1-\gamma_s}{1-\gamma_s^n}\left(R_{22}[n] + \widehat{\boldsymbol{\alpha}}^{\mathsf{T}}[n+1]R_{12}[n]\right) \tag{2.133}$$

$$\widehat{g}_w[n+1] = \frac{1-\gamma_w}{1-\gamma_w^n}Q_{11}[n] \tag{2.134}$$

$$\widehat{\boldsymbol{a}}[n+1] = A_{11}^{-1}[n]A_{12}[n]$$

$$\widehat{g}_1[n+1] = \frac{1-\gamma_a}{1-\gamma_a^n}\left(A_{22}[n] - \widehat{\boldsymbol{a}}^{\mathsf{T}}[n+1]A_{12}[n]\right) \tag{2.135}$$

$$\widehat{\boldsymbol{b}}[n+1] = B_{11}^{-1}[n]B_{12}[n]$$

$$\widehat{g}_2[n] = \frac{1-\gamma_b}{1-\gamma_b^n}\left(B_{22}[n] - \widehat{\boldsymbol{b}}^{\mathsf{T}}[n+1]B_{12}[n]\right). \tag{2.136}$$

Note that the recursive updates themselves require no matrix inverses, though the actual parameter estimates do. This can be exploited for computational savings by updating the recursively computed quantities with each new data sample, but only intermittently using these quantities to calculate new parameter estimates. In actual experiments, the parameter estimates vary slowly enough that this strategy can be employed without significant loss of accuracy.

As a final observation, note that $\boldsymbol{\alpha}$, $\boldsymbol{a}$, and $\boldsymbol{b}$ should be computed before the other parameters, as the other parameter estimate equations depend on these quantities.

The Sequential Form of the Gradient-Based M-step

The gradient-based parameter estimation equations can also be converted to be causal and recursively-computable using the same strategy as used in the previous section for the exact solution parameter estimation equations. Again, iteration indices are replaced with time indices. However, in this version of the M-step, the sums over the entire observation interval are replaced with the current estimate of the summand, rather than the weighted sums used above. Following this strategy gives

$$\begin{aligned}
\widehat{\boldsymbol{\alpha}}[n+1] &= \widehat{\boldsymbol{\alpha}}[n] - \frac{\delta_\alpha}{\widehat{g}_s[n]} \cdot \Big[\widehat{\boldsymbol{s}_{p-1}[n-1|n]s[n|n]} \\
&\qquad + \widehat{\boldsymbol{s}_{p-1}[n-1|n]\boldsymbol{s}_{p-1}^{\mathsf{T}}[n-1|n]}\,\widehat{\boldsymbol{\alpha}}[n]\Big] &(2.137)\\
\widehat{g}_s[n+1] &= \left(1-\frac{\tilde{\delta}_s}{2}\right)\widehat{g}_s[n] + \frac{\tilde{\delta}_s}{2}\cdot\Big[\widehat{s^2}[n|n] + 2\widehat{\boldsymbol{\alpha}}^{\mathsf{T}}[n]\widehat{\boldsymbol{s}_{p-1}[n-1|n]s[n|n]} \\
&\qquad + \widehat{\boldsymbol{\alpha}}^{\mathsf{T}}[n]\widehat{\boldsymbol{s}_{p-1}[n-1|n]\boldsymbol{s}_{p-1}^{\mathsf{T}}[n-1|n]}\,\widehat{\boldsymbol{\alpha}}[n]\Big] &(2.138)\\
\widehat{g}_w[n+1] &= \left(1-\frac{\tilde{\delta}_w}{2}\right)\widehat{g}_w[n] + \frac{\tilde{\delta}_w}{2}\widehat{w^2}[n|n] &(2.139)\\
\widehat{\boldsymbol{a}}[n+1] &= \widehat{\boldsymbol{a}}[n] + \frac{\delta_a}{\widehat{g}_1[n]}\Big[\widehat{\boldsymbol{w}}_q[n|n]z_1[n] - \widehat{\boldsymbol{w}_q[n|n]s[n|n]} \\
&\qquad - \widehat{\boldsymbol{w}_q[n|n]\boldsymbol{w}_q^{\mathsf{T}}[n|n]}\,\widehat{\boldsymbol{a}}[n]\Big] &(2.140)\\
\widehat{g}_1[n+1] &= \left(1-\frac{\tilde{\delta}_1}{2}\right)\widehat{g}_1[n] + \frac{\tilde{\delta}_1}{2}\Big[z_1^2[n] - 2z_1[n]\widehat{s}[n|n] + \widehat{s^2}[n|n] \\
&\qquad - 2\widehat{\boldsymbol{a}}^{\mathsf{T}}[n]\widehat{\boldsymbol{w}}_q[n|n]z_1[n] + 2\widehat{\boldsymbol{a}}^{\mathsf{T}}[n]\widehat{\boldsymbol{w}_q[n|n]s[n|n]} \\
&\qquad + \widehat{\boldsymbol{a}}^{\mathsf{T}}[n]\widehat{\boldsymbol{w}_q[n|n]\boldsymbol{w}_q^{\mathsf{T}}[n|n]}\,\widehat{\boldsymbol{a}}[n]\Big] &(2.141)\\
\widehat{\boldsymbol{b}}[n+1] &= \widehat{\boldsymbol{b}}[n] + \frac{\delta_b}{\widehat{g}_2[n]}\Big[\widehat{\boldsymbol{s}}_r[n|n]z_2[n] - \widehat{\boldsymbol{s}_r[n|n]w[n|n]} - \widehat{\boldsymbol{s}_r[n|n]\boldsymbol{s}_r^{\mathsf{T}}[n|n]}\,\widehat{\boldsymbol{b}}[n]\Big] &(2.142)\\
\widehat{g}_2[n+1] &= \left(1-\frac{\tilde{\delta}_2}{2}\right)\widehat{g}_2[n] + \frac{\tilde{\delta}_2}{2}\Big[z_2^2[n] - 2z_2[n]\widehat{w}[n|n] + \widehat{w^2}[n|n] \\
&\qquad - 2\widehat{\boldsymbol{b}}^{\mathsf{T}}[n]\widehat{\boldsymbol{s}}_r[n|n]z_2[n] + 2\widehat{\boldsymbol{b}}^{\mathsf{T}}[n]\widehat{\boldsymbol{s}_r[n|n]w[n]} \\
&\qquad + \widehat{\boldsymbol{b}}^{\mathsf{T}}[n]\widehat{\boldsymbol{s}_r[n|n]\boldsymbol{s}_r^{\mathsf{T}}[n|n]}\,\widehat{\boldsymbol{b}}[n]\Big]. &(2.143)
\end{aligned}$$

Chapter 3

Low Order Simulation

This chapter describes a simulation using synthetically generated data which conforms to the assumptions made by the algorithm. This experiment also provides a rough benchmark or calibration point indicating how well the algorithm can be expected to perform in a "best-case" situation.

3.1 Simulation Results

The desired signal was generated by a second order autoregressive process driven by white noise. The poles of the AR filter were at $z = 0.9e^{\pm j\pi/4}$, and the driving noise had a variance of 100. The resulting desired signal $s[n]$ is shown in Figure 3-1. The figure shows the samples of the discrete-time data connected by straight lines. The coupling filters were ninth order FIR filters. The impulse responses $a[n]$ and $b[n]$ were samples of exponentially-damped sinusoids. The impulse responses and their corresponding frequency responses are shown in Figures 3-2 through 3-5.

The corrupting noise $w[n]$ was white, Gaussian, noise independent of the noise driving the AR filter. The noise process generating $w[n]$ was zero-mean with a variance of 1984. This value of the variance assured that the signal-to-noise ratio measured at $z_1[n]$ was 0 dB. This SNR value was obtained using the following formula:

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{n=1}^{N}(s[n])^2}{\sum_{n=1}^{N}(z_1[n]-s[n])^2}\right). \tag{3.1}$$

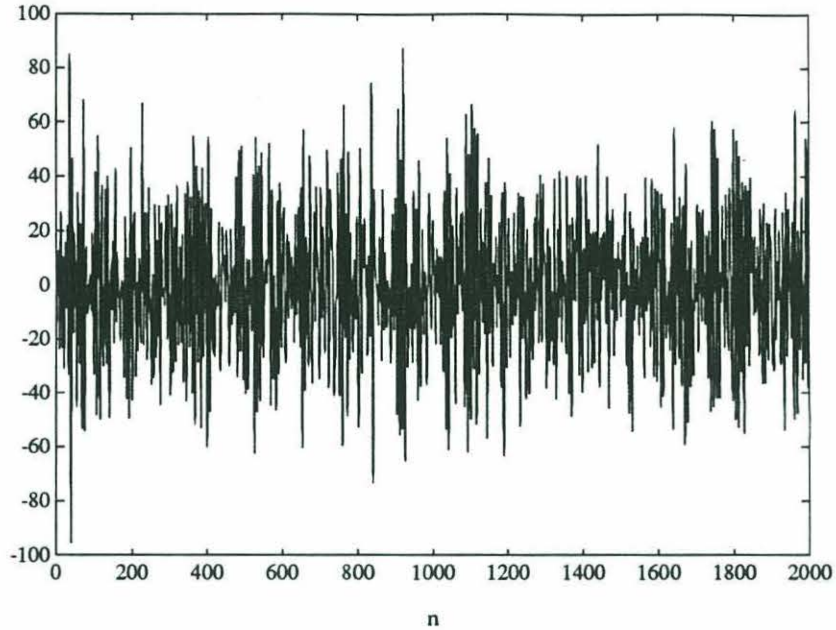

Figure 3-1: Desired Signal $s[n]$

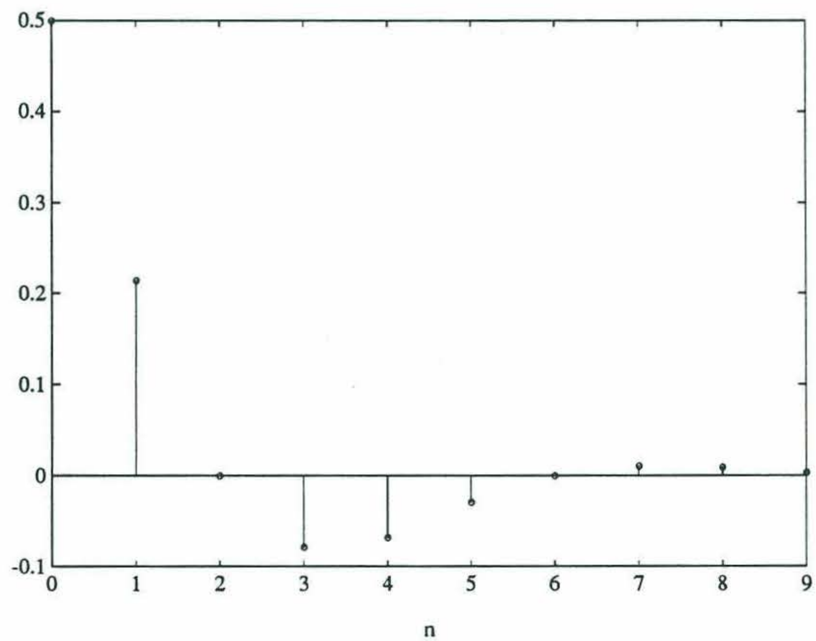

Figure 3-2: Impulse Response $a[n]$

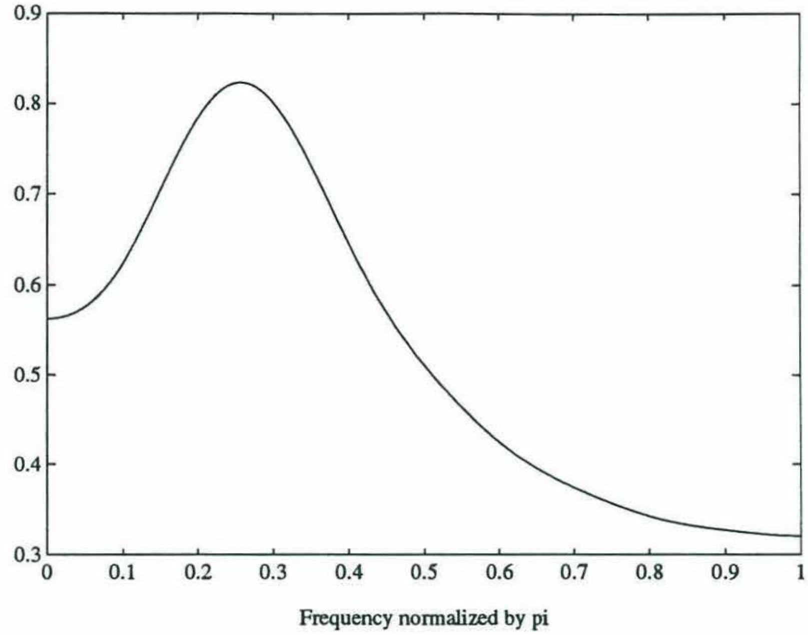

Figure 3-3: Frequency Response $|A(e^{j\omega})|$

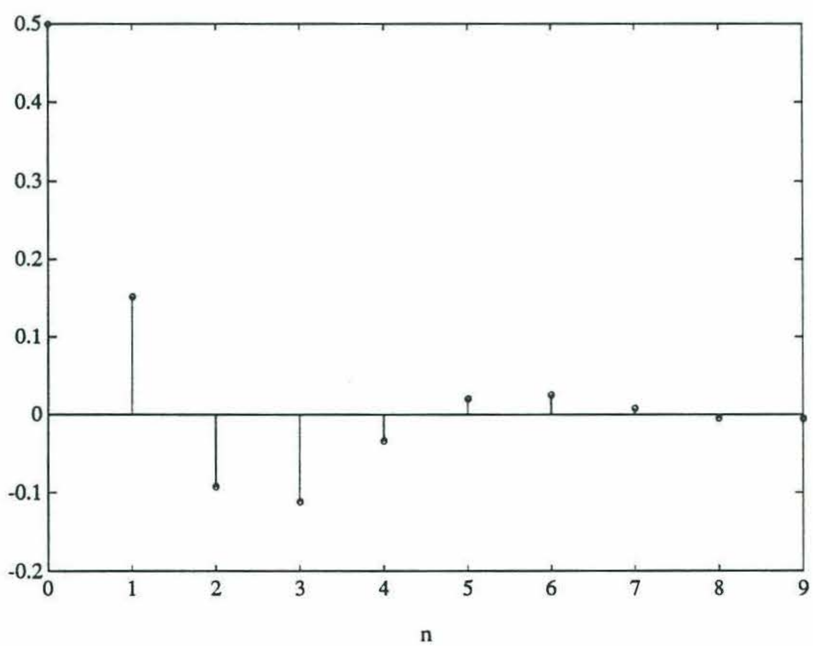

Figure 3-4: Impulse Response $b[n]$

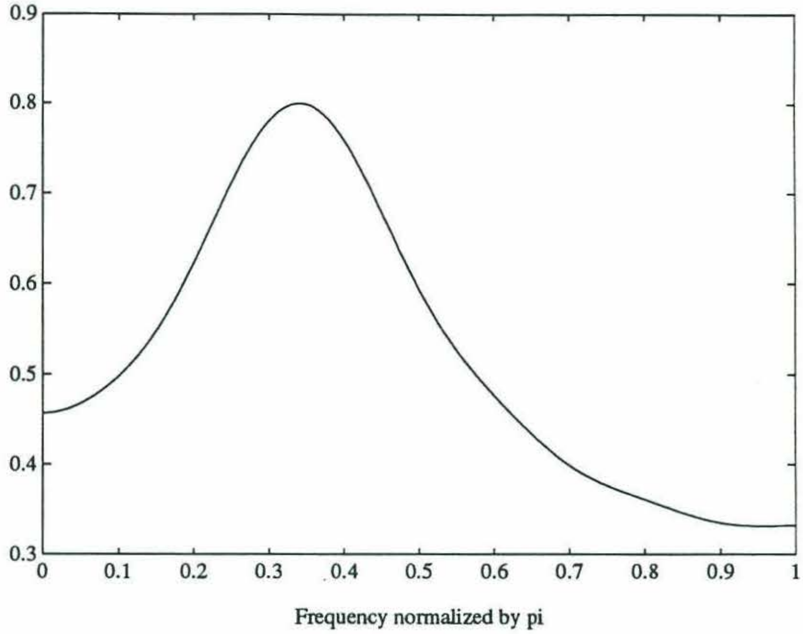

Figure 3-5: Frequency Response $|B(e^{j\omega})|$

The SNR measured at the secondary sensor, $z_2[n]$, was roughly -6 dB. This value was obtained by using the formula in Eq. (3.1), except $z_1[n]$ was replaced by $z_2[n]$.

The incremental noise sources $e_1[n]$ and $e_2[n]$ were Gaussian, white, independent, and zero-mean with variance 0.0625. The sensor values shown in Figures 3-6 and 3-7 were created using these signals and Eqs. (2.19) and (2.20). These discrete-time samples were also connected by straight lines to avoid overly cluttered plots.

Though it is not explicitly mentioned in [TR 532], Feder found that his noncausal algorithm did not converge if both the coupling filters were estimated. Preliminary investigations on the algorithm described in the last chapter encountered the same problem. Consequently, the impulse response of only one of the coupling filters $a[n]$, the AR coefficients $\alpha[n]$, and variance of the driving noise, g_s, were estimated in the experiments described in this thesis. The impulse response $b[n]$, and the variances g_w, g_1, and g_2, were assumed to be known and held fixed.

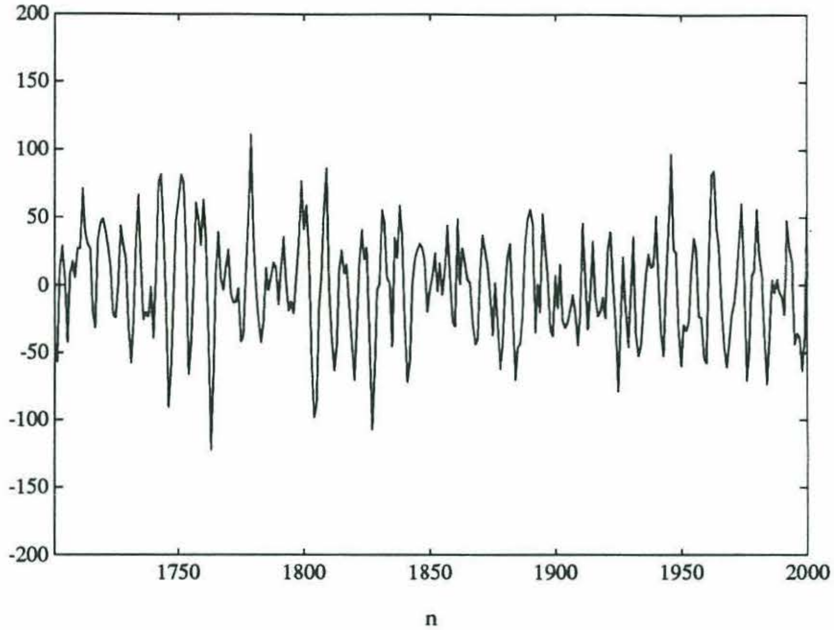

Figure 3-6: Primary Sensor $z_1[n]$

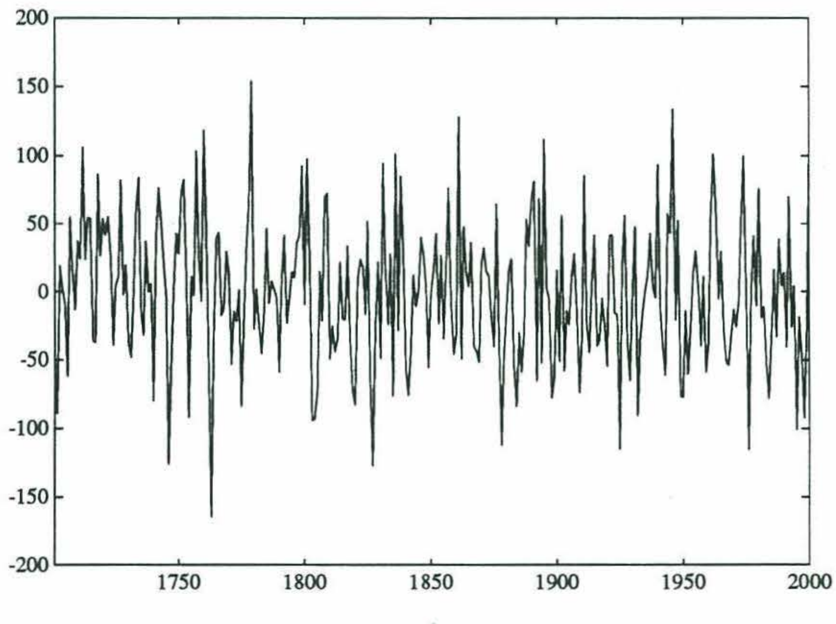

Figure 3-7: Secondary Sensor $z_2[n]$

Preliminary investigations also revealed that the algorithm adapted very slowly to the values of $a[n]$ if it had no prior information, i.e., if all the coefficients initially started at zero. Again, though not mentioned explicitly, this difficulty was also encountered in the work done for [TR 532]. To avoid this problem the coefficients were initialized with a least-squares estimate based on an initial block of the sensor data using the system model shown in Figure 1-1. This gave an initial estimate of the value of $a[n]$ in the rough neighborhood of the true value.

The final issue addressed in preparing for the simulation was the initial conditions for the state vector and covariance matrix for the Kalman filter. The EM algorithm was started from sample n_0, where $n_0 = \max(q, r) + 2$. The top part of the state vector, $\boldsymbol{s}_r[n_0 - 1]$, was initially set to be $z_1[n_0 - r - 1], \ldots, z_1[n_0 - 1]$, and similarly $\boldsymbol{w}_q[n_0 - 1]$ was set to be $z_2[n_0 - q - 1], \ldots, z_2[n_0 - 1]$. The first $q + 1$ diagonal elements of the covariance matrix were set to be the initial estimate of g_s, and the remaining $r + 1$ diagonal elements were set to be g_w. All state variables were initially assumed to be uncorrelated, so all the off-diagonal elements of the covariance matrix were started at zero. It was assumed that crude a priori knowledge about g_s was available. When the precise solution M-step was used, the initial guess of g_s didn't need to be particularly accurate, as the estimate adapted quite nicely.

Running the algorithm on the data from Section 3.1 resulted in approximately 19 dB enhancement in terms of signal-to-error ratio (SER). Signal-to-error ratio was calculated using the following formula:

$$\mathrm{SER} = 10 \log_{10} \left(\frac{\sum_{n=n_0}^{N} (s[n])^2}{\sum_{n=n_0}^{N} (s[n] - \widehat{s}[n])^2} \right), \tag{3.2}$$

where $\widehat{s}[n]$ was the estimate produced by the algorithm of the desired signal. As shown in Figure 3-8, this estimated signal tracked the true signal closely.

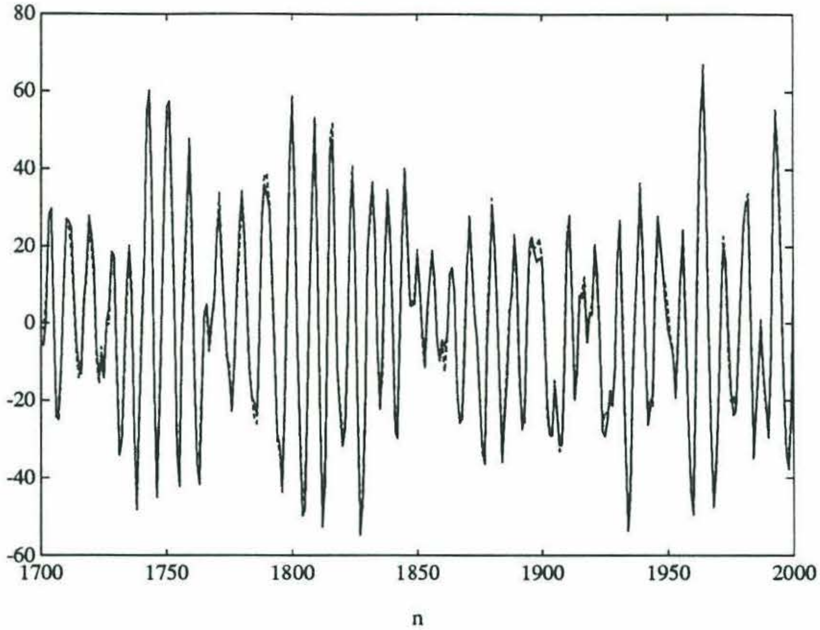

Figure 3-8: Actual(solid) and Estimated(dashed) AR Signal

As noted earlier, the EM algorithm is intended for parameter estimation. The accurate state estimates in the E-step should result in accurate parameter estimates, too. Figure 3-9 shows the close fit between the estimated filter coefficients and the actual values. Figure 3-10 plots the time profile of the estimate of g_s, and Figure 3-11 shows the time profile of the estimates of the AR parameters of the driving process. In the latter two plots, the solid lines indicate the actual values, and although there was some wavering, it can be seen the estimates ultimately converged on the true values. In the case of the AR parameters, the convergence was fairly fast. Once again, both these graphs actually plot the discrete samples connected by straight lines.

Given the evidence that the algorithm worked under favorable conditions, it was next evaluated on a more challenging and realistic data set as described in the next chapter.

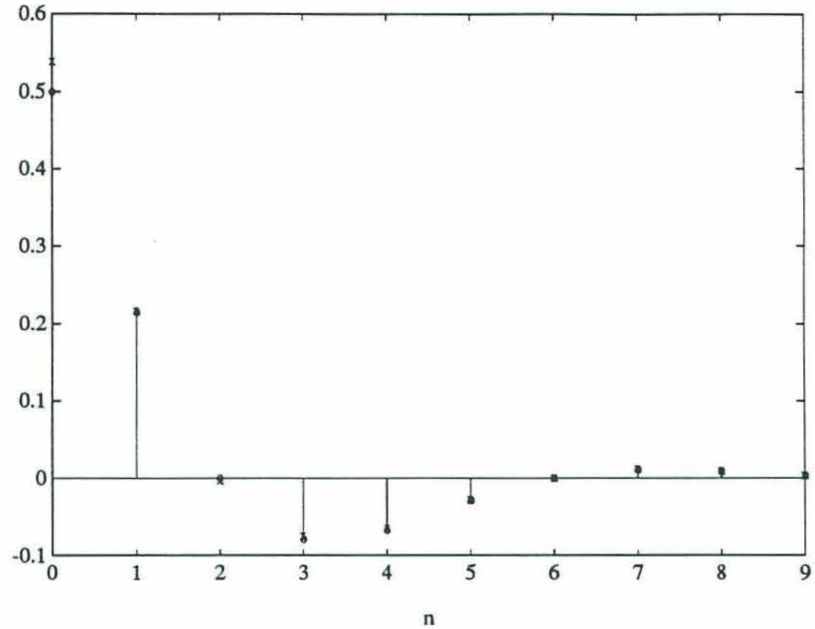

Figure 3-9: Actual(o) and Estimated(x) Impulse Response $a[n]$

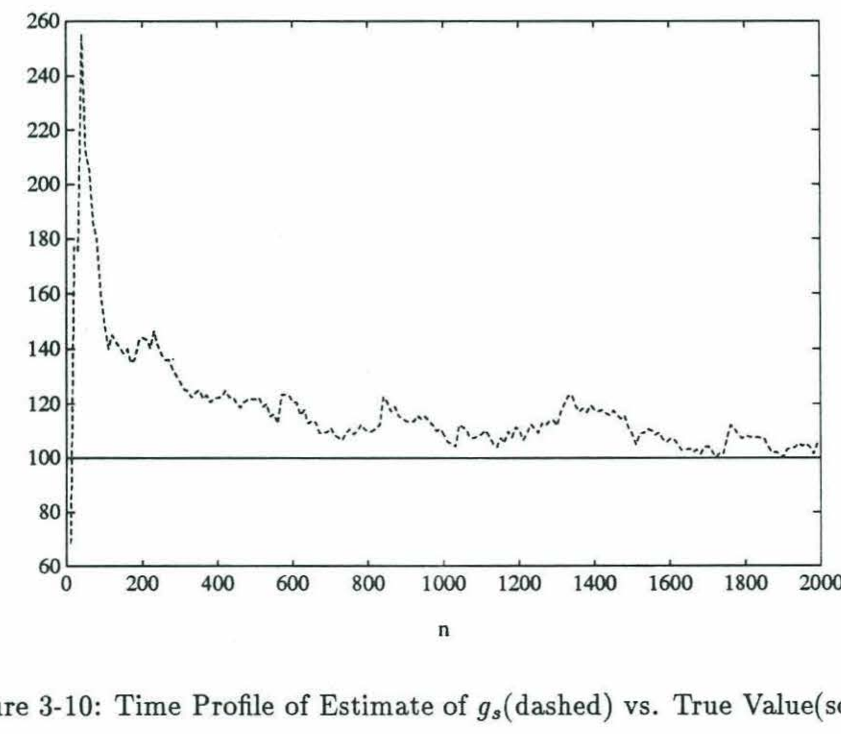

Figure 3-10: Time Profile of Estimate of g_s(dashed) vs. True Value(solid)

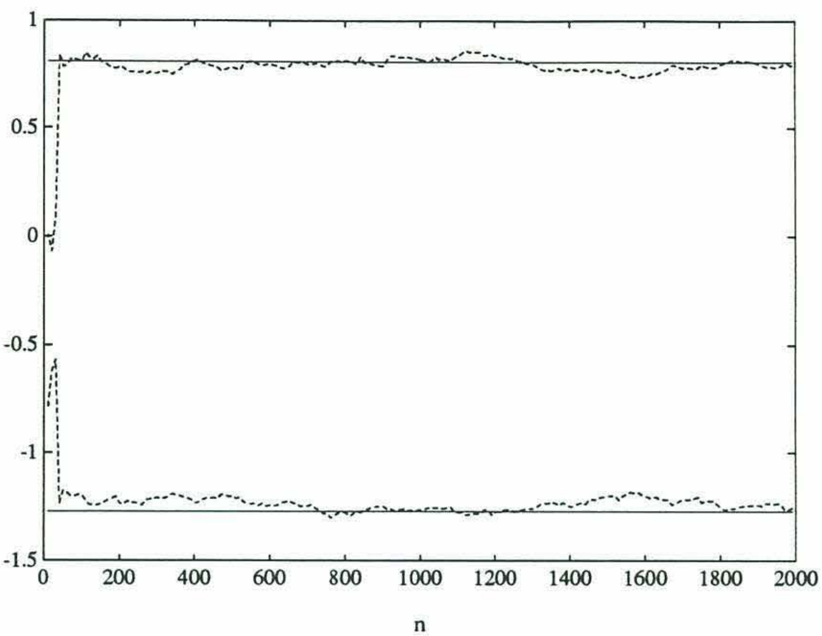

Figure 3-11: Time Profile of Estimate of coefficients $\alpha[n]$(dashed) vs. True Values(solid)

Chapter 4

Speech in Simulated Room Acoustics

This chapter describes a more interesting and potentially more practical application of the signal enhancement algorithm. In this experiment speech data corrupted with white noise was used. The coupling filters were simulated room impulse responses. The performance of the algorithm described in Chapter 2 is compared against the performance of the noncausal algorithm described in [TR 532] using the same data.

4.1 Simulation Results

For this experiment, the desired signal $s[n]$ was the spoken sentence "He has the bluest eyes." A plot of this signal is shown in Figure 4-1. In all the plots in this chapter and the next, discrete-time samples are connected by straight line to avoid undue clutter in the figures. The corrupting noise $w[n]$ was white, independent, and Gaussian. In addition, g_w was set such that the SNR measured at $z_1[n]$ was 0 dB. The SNR at the secondary sensor $z_2[n]$ fell far below that, at roughly -26 dB SNR.

The cross-coupling filters A and B were derived from the simulated room acoustics impulse responses used in [TR 532]. The experiment described in [TR 532] used 255^{th} order FIR filters, but for computational reasons the experiment described here used 127^{th} order filters derived from the 255^{th} order filters by frequency sampling. This gave a close approximation to the same frequency response for the room acoustics. The impulse responses $a[n]$ and $b[n]$, along with their frequency responses, are shown in Figures 4-2 through 4-5.

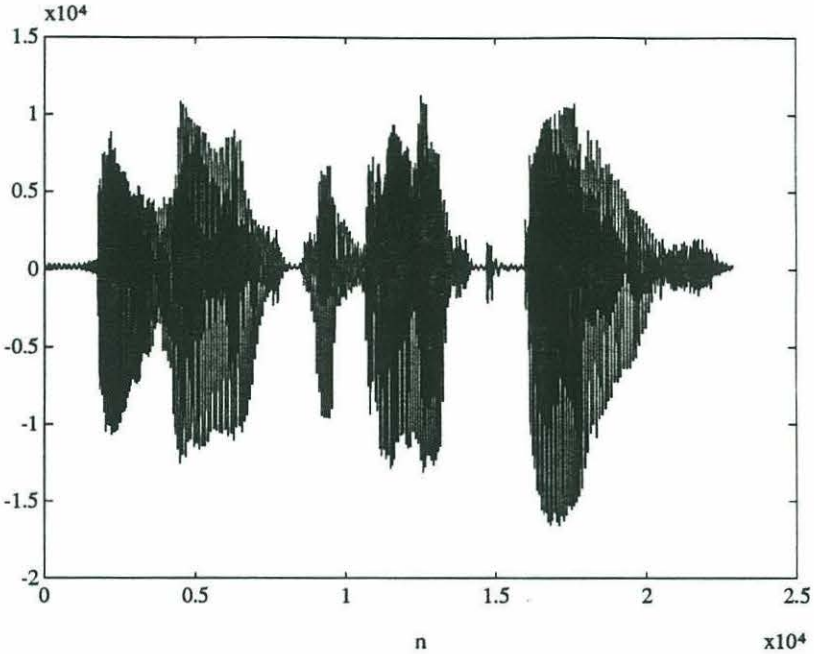

Figure 4-1: $s[n]$: "He has the bluest eyes."

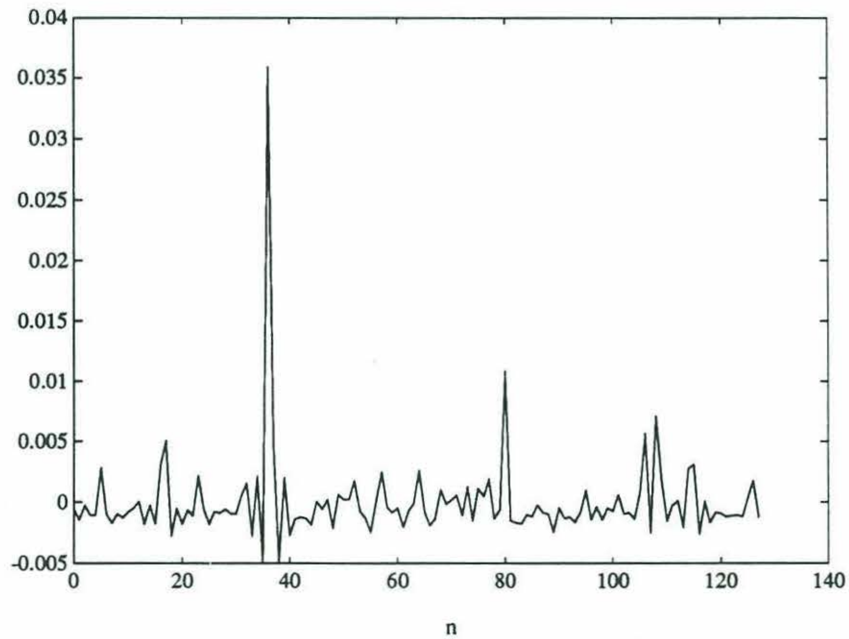

Figure 4-2: Impulse Response $a[n]$

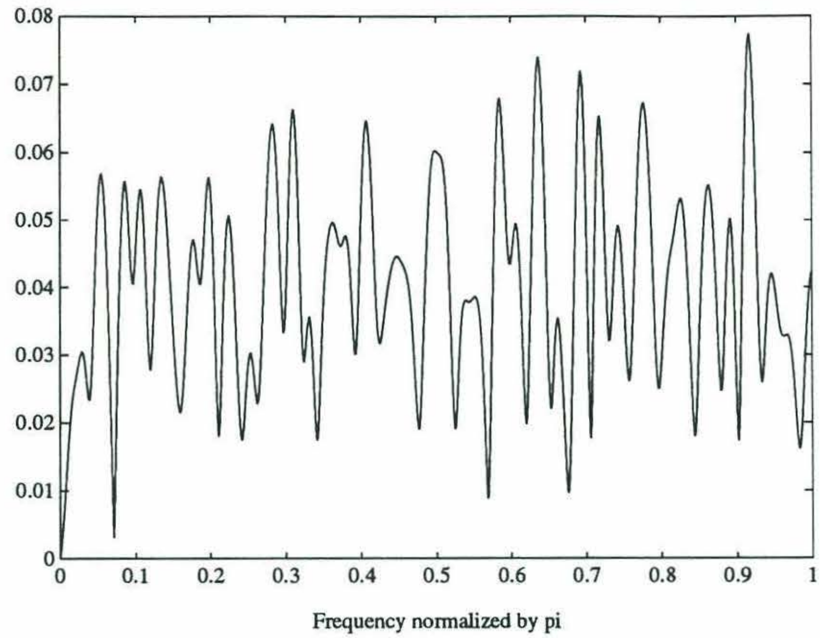

Figure 4-3: Frequency Response $|A(e^{j\omega})|$

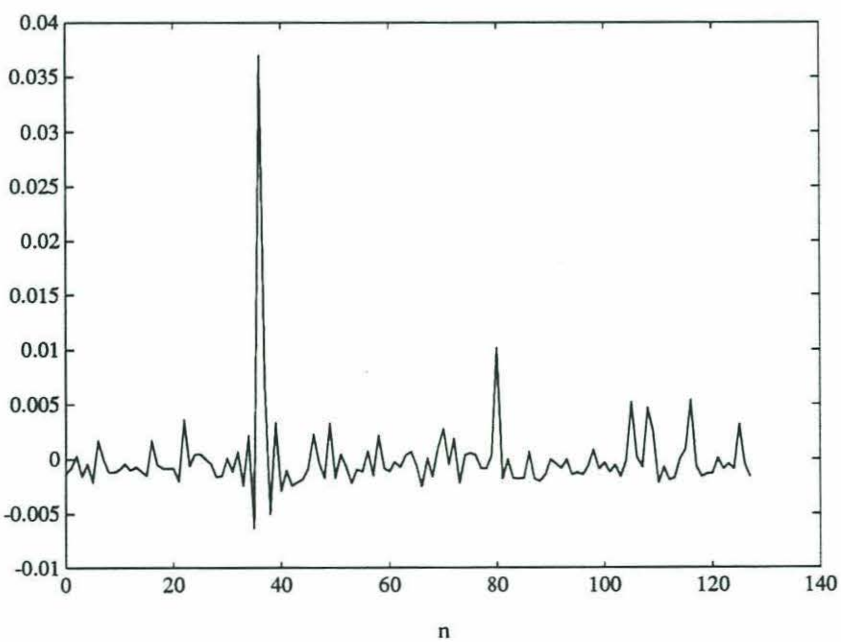

Figure 4-4: Impulse Response $b[n]$

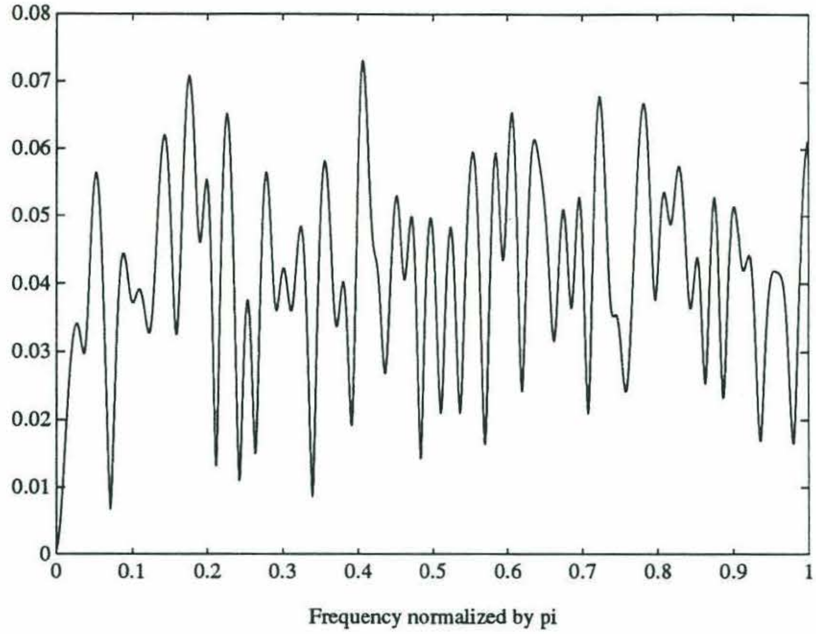

Figure 4-5: Frequency Response $|B(e^{j\omega})|$

The variances g_1 and g_2 were set to be 100, far below g_s and g_w, and $e_1[n]$ and $e_2[n]$ were white, Gaussian, and independent of all other noise processes. The synthetic sensor data $z_1[n]$ and $z_2[n]$ were generated according to Eqs. (2.17) and (2.18). One segment of the data is shown in Figures 4-6 and 4-7. Figure 4-8 shows the original uncorrupted speech for the same segment, as a basis for comparison.

The experiment essentially followed the same procedure described in Section 3.1. The initial state vector values were taken directly from the sensors, and the initial covariance matrix values were set to be zero off the diagonal and g_s and g_w on the diagonal. The only parameters actually estimated were the coefficients of $a[n]$, the AR parameters $\alpha[n]$, and g_s. All the other parameters were assumed to be known and held fixed. The exact solution parameter estimates were used for the M-step, and the estimate of $a[n]$ was initialized using a least-squares fit.

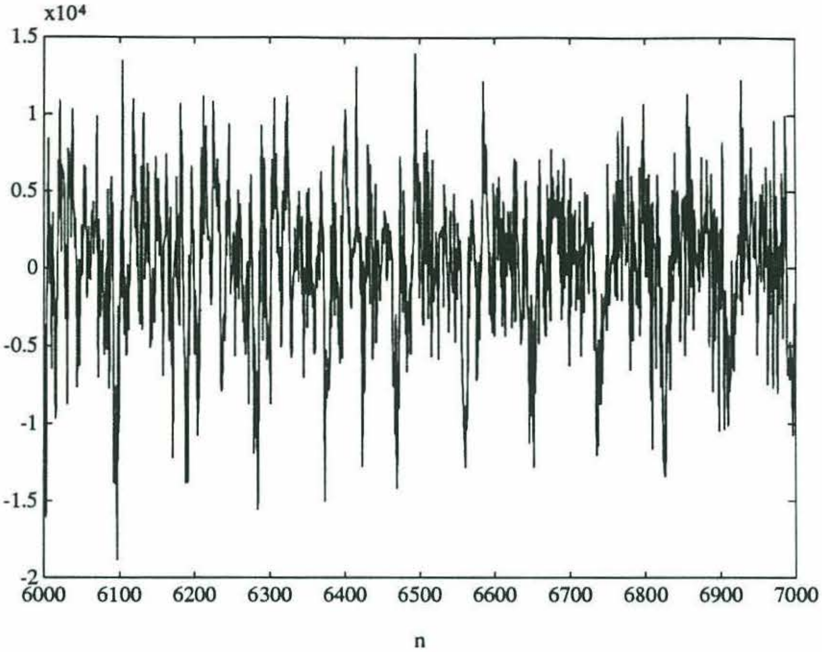

Figure 4-6: Primary Sensor $z_1[n]$

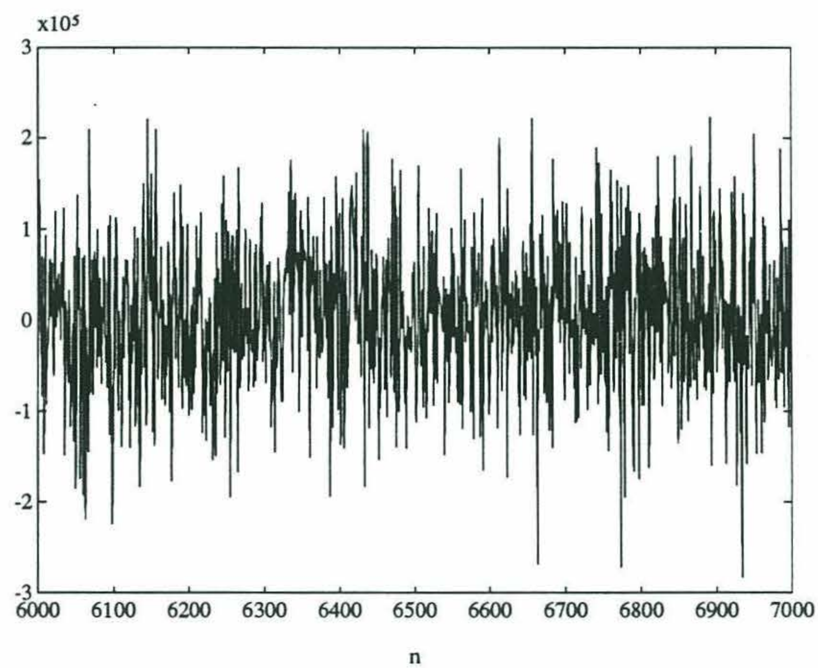

Figure 4-7: Secondary Sensor $z_2[n]$

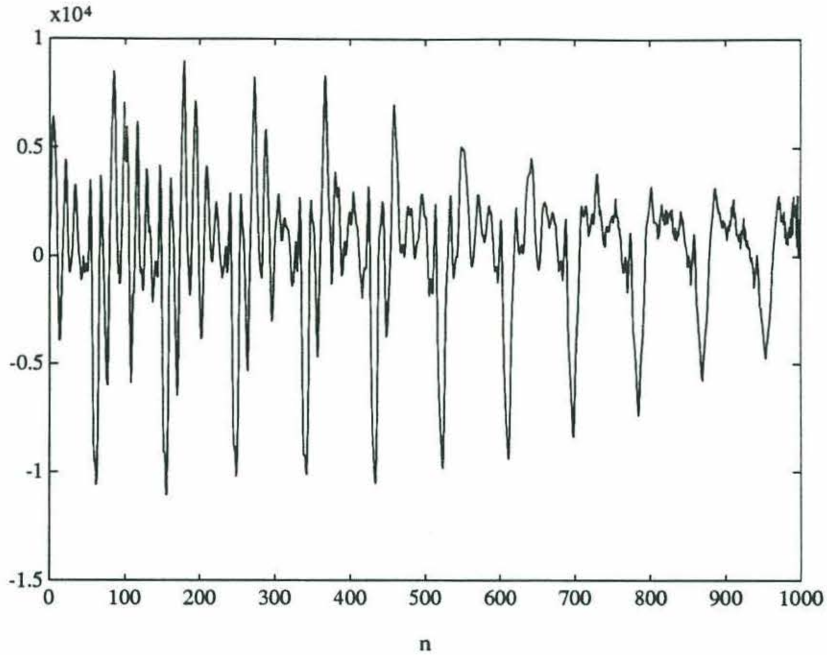

Figure 4-8: Original Signal $s[n]$

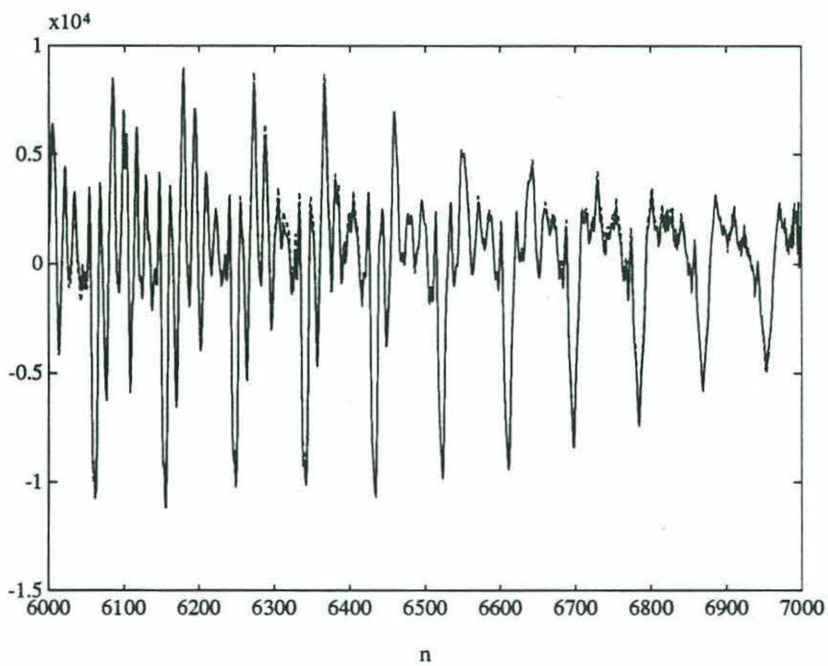

Figure 4-9: Actual(solid) and Estimated(dashed) Speech Signal $s[n]$

Using actual speech as the desired signal $s[n]$ added an additional consideration, as it was no longer known a priori what order autoregressive model to use for $s[n]$. In general, speech is thought to be well-modeled by a 10^{th} to 14^{th} order AR process [RS 78]. However, preliminary experimentation with various orders in this algorithm indicated high-order AR models such as these tended to generate unstable or inaccurate estimates. The work presented in [TR 532] also encountered this difficulty. Because of this, a second-order AR process was used for the model of $s[n]$, i.e., $p = 2$. Although this is gross under-parameterization of $s[n]$, the algorithm still performed significant enhancement. This implies that the essential requirement for good enhancement is an accurate estimate of the impulse response $a[n]$, and the model for $s[n]$ is much less important.

Applying the algorithm to the data created as described above resulted in about 23 dB of enhancement in terms of SER, calculated using Eq. (3.2). As shown in Figure 4-9, the enhanced signal tracks the actual speech signal for the same segment shown earlier in Figures 4-6 and 4-7, even during the transition in the signal at the end of the segment shown. Figures 4-12 and 4-13 show the error and speech signal plotted adjacent to each other. There appears to be no gross correlation between the error and the signal being estimated.

The algorithm also generated an accurate estimate of the filter coefficients. Figure 4-10 shows the estimated impulse response of $a[n]$ plotted against the actual impulse response used to create the data, while Figure 4-11 shows the frequency response magnitude $|A(e^{j\omega})|$ for both the estimated and actual filter. In both graphs, it can be seen that the estimate agrees closely with the true value in most places. The estimate of the driving variance for g_s also appears to work well. Figure 4-14 shows a time profile of the estimated value of g_s, with the actual signal $s[n]$ plotted directly below on the same time scale. It is clear that the estimate of g_s tracks the envelope of the speech accurately.

In addition to the absolute performance measurements for this experiment, the performance of the algorithm relative to the algorithm described in [TR 532] is of interest, as this will give some indication of the performance cost paid for using a causal, sequential algorithm, contrasted with a noncausal, iterative, block algorithm. Running the noncausal algorithm of [TR 532] on the same data set described earlier in this chapter gave 28 dB enhancement, calculated using the equation for SER given by Eq. (3.2). Consequently, 5 dB of performance was sacrificed in making the algorithm causal and sequential.

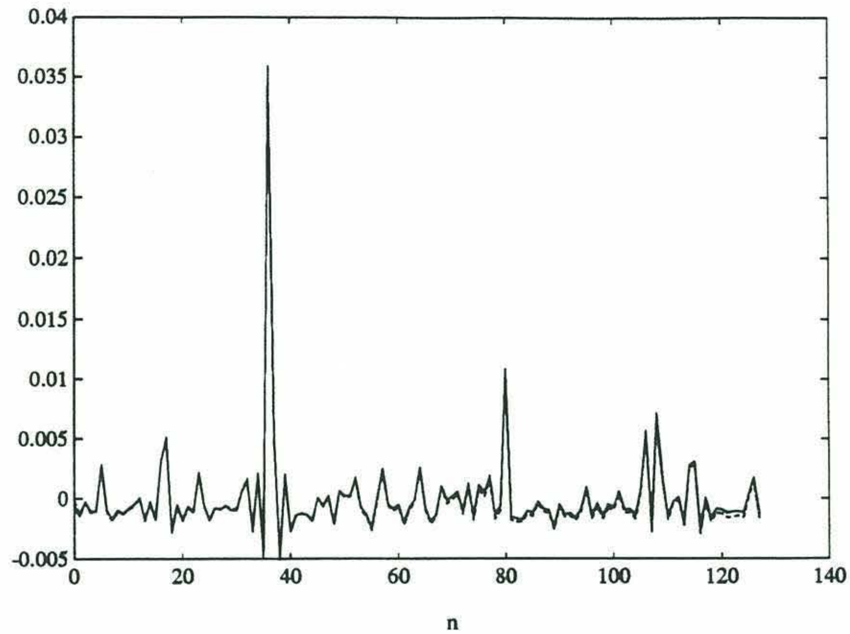

Figure 4-10: Actual(solid) and Estimated(dashed) $a[n]$

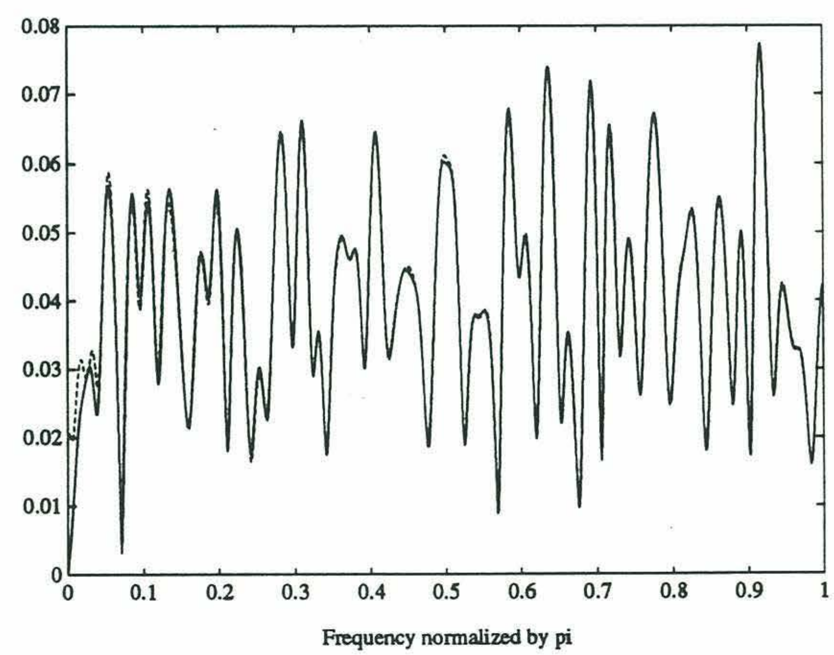

Figure 4-11: Actual(solid) and Estimated(dashed) $|A(e^{j\omega})|$

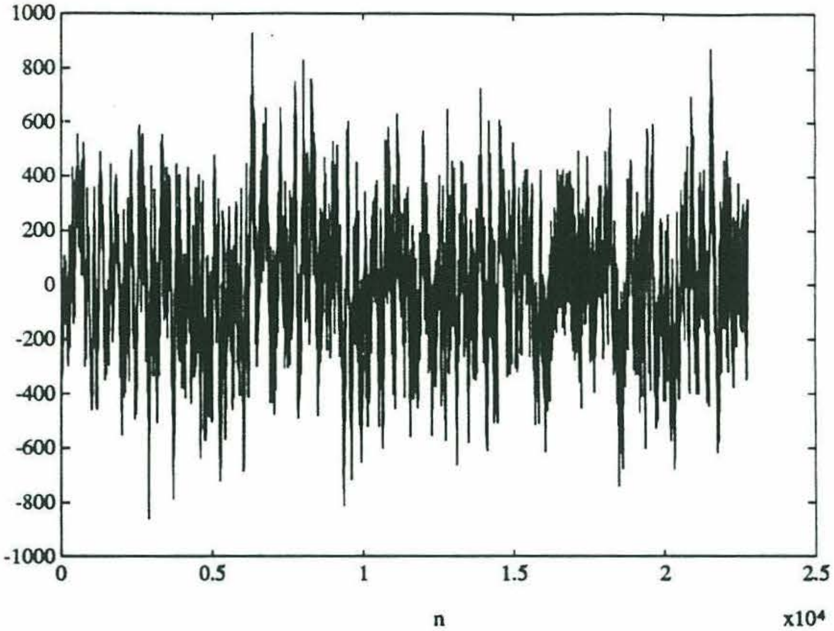

Figure 4-12: Error Signal

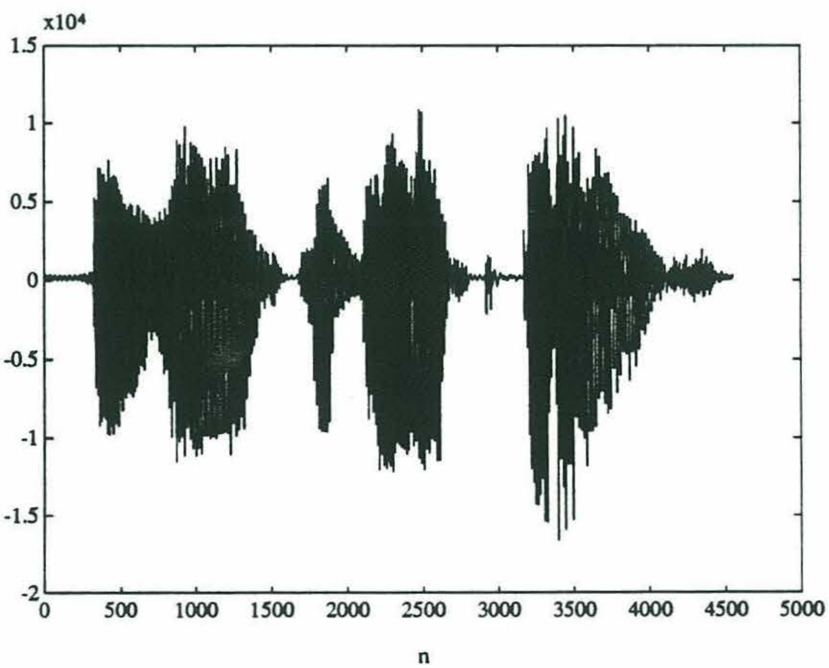

Figure 4-13: Speech Signal $s[n]$

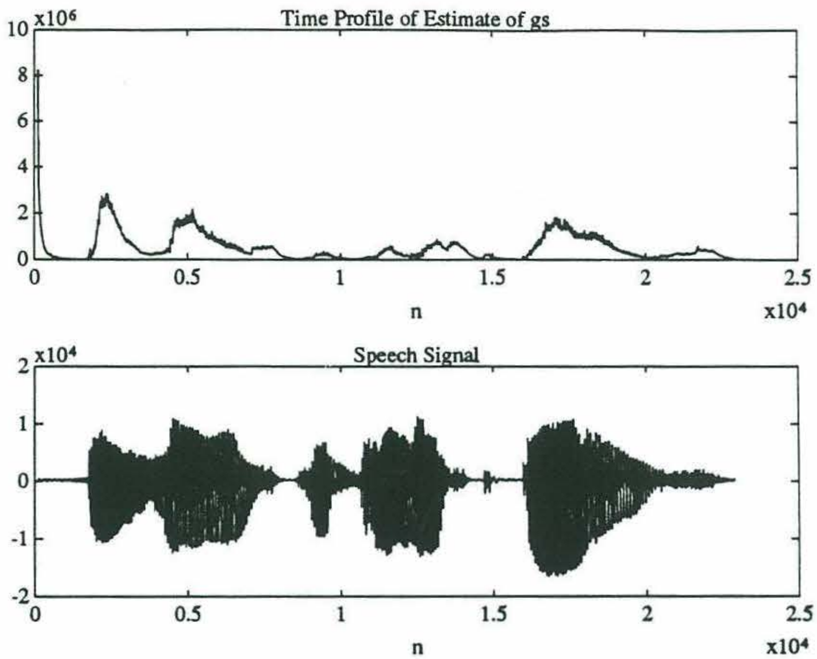

Figure 4-14: Time Profile of Estimate of g_s

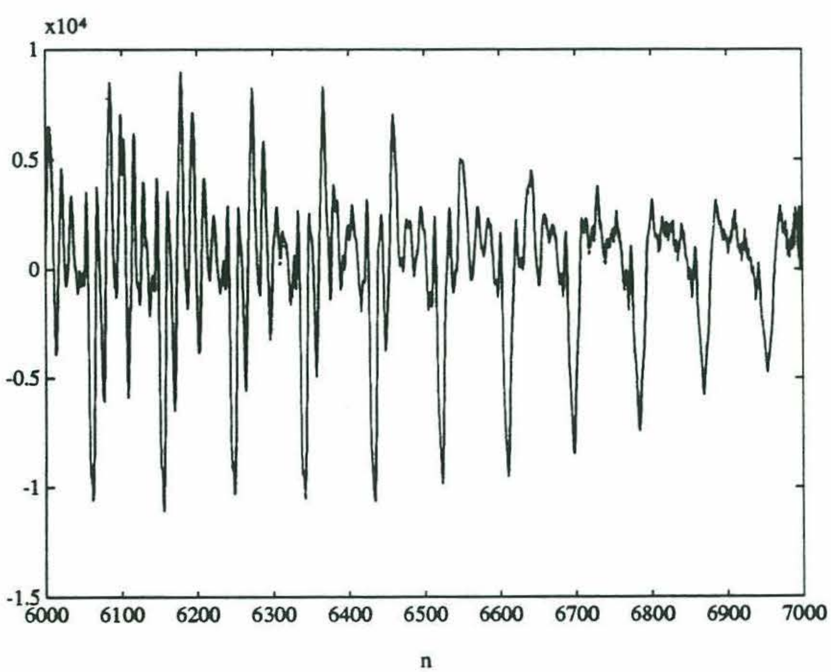

Figure 4-15: Feder's Estimated(dashed) and the Actual(solid) Signal $s[n]$

Figure 4-15 shows the signal estimate generated by the noncausal algorithm, along with the actual signal value for the same time segment as shown in Figures 4-6 and 4-7. Again, the estimate is very accurate, as the SER measurement indicated it would be.

While the causal algorithm does not equal the performance of the noncausal algorithm, it does attain a high level of enhancement. The computational power required to run either algorithm in real time greatly exceeds that currently commonly available. For this reason, modifications of the algorithm reducing the computational complexity are very desirable, so long as they still provide a satisfactory level of enhancement. The next chapter describes an experiment examining the performance of the gradient-based algorithm, which is computationally simpler that either of the algorithms examined in this chapter.

Chapter 5

Speech Enhancement Using the Gradient-Based Algorithm

This chapter describes an experiment focusing on the performance tradeoff involved in using the gradient-based M-step on the data set from the last chapter. The fact that this algorithm is computationally cheaper and simpler makes it attractive, but it is important to understand the performance cost paid for these benefits.

5.1 Simulation Results

The experiment described in this chapter used exactly the same data set as in the previous chapter, i.e., the speech signal "He has the bluest eyes" was the desired signal, the corrupting noise was white and Gaussian, and the coupling filters were the simulated room impulse responses.

Once again, only $\boldsymbol{a}$, $\boldsymbol{\alpha}$, and g_s were estimated. The Kalman state vector was started with sensor values, and the covariance matrix with zeros expect for initial variance estimates on the diagonal. However, the initial estimates for $\boldsymbol{a}$ were not a least-squares fit, but instead zero, indicating no prior knowledge about the coefficient values. Lastly, $s[n]$ was again modeled using a second order AR process.

For 127^{th} order filters, the gradient-based M-step only required 2% of the floating point operations that the precise solution M-step needed.

The gradient-based M-step adapted to much more accurate estimates of the filter coefficients than the precise solution M-step did without prior information. Starting from zero, as

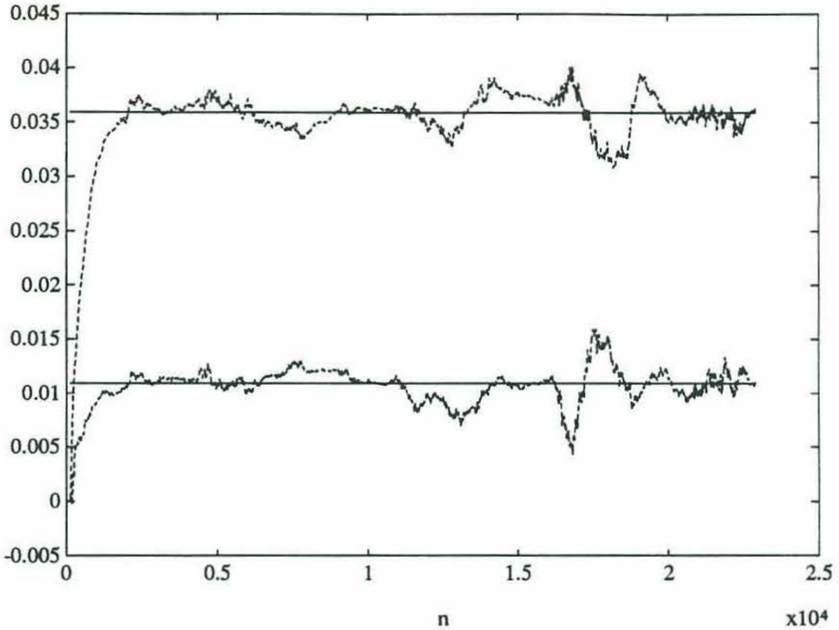

Figure 5-1: Time Profile of Prominent Coefficients of $a[n]$

noted above, it obtained accurate estimates in roughly 2000 data points, which corresponds to about one-fifth of a second for the 10KHz sampling frequency used for the data. Figure 5-1 shows the time profile of the estimates for the two largest coefficients. In this figure, it can be seen that the estimates converged to the neighborhood of their true values fairly quickly. The solid lines indicate the true values, while the dashed lines are the estimated values. The gradient-based M-step produced accurate final estimates of the whole impulse response as well, as shown in Figure 5-2

Looking at the actual signal estimates, they are not as accurate as those given by the M-step that precisely solved the estimated likelihood function. Figure 5-3 shows the signal estimate of the gradient-based algorithm for the same segment shown in Figure 4-9. While the estimate generally had roughly the right shape, it had difficulty adapting during the transition at the end of the segment. Figure 5-4 plots the error signal and speech signal. Unlike Figure 4-12, the error tended to grow larger during transition periods of the signal. Using Eq. (3.2) to calculate the overall signal-to-error ratio gives a value of 8 dB. However, this number is slightly misleading. Consider Figure 5-5, which shows a time profile of SER of enhancement. The value at any time point is the SER calculated over the last 100 samples

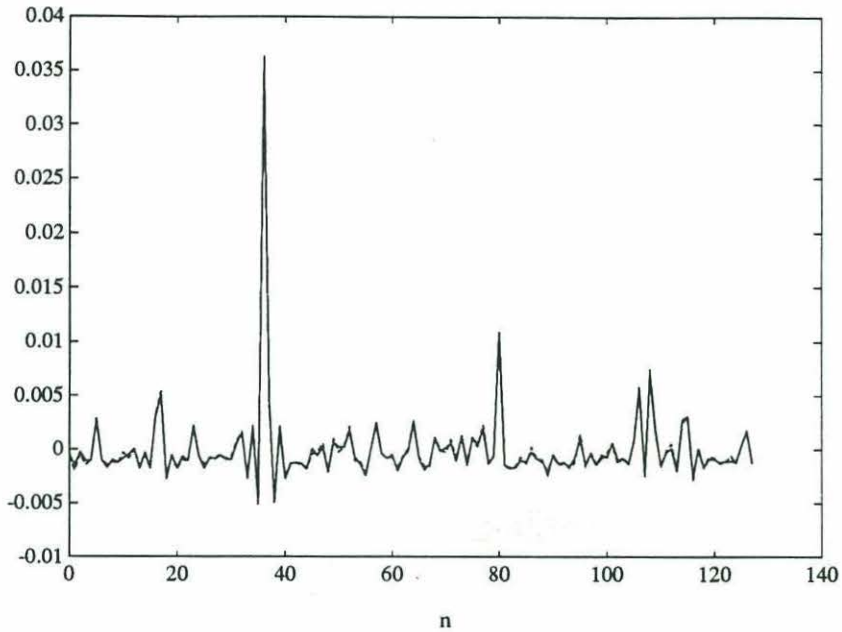

Figure 5-2: Actual(solid) and Estimated(dashed) $a[n]$

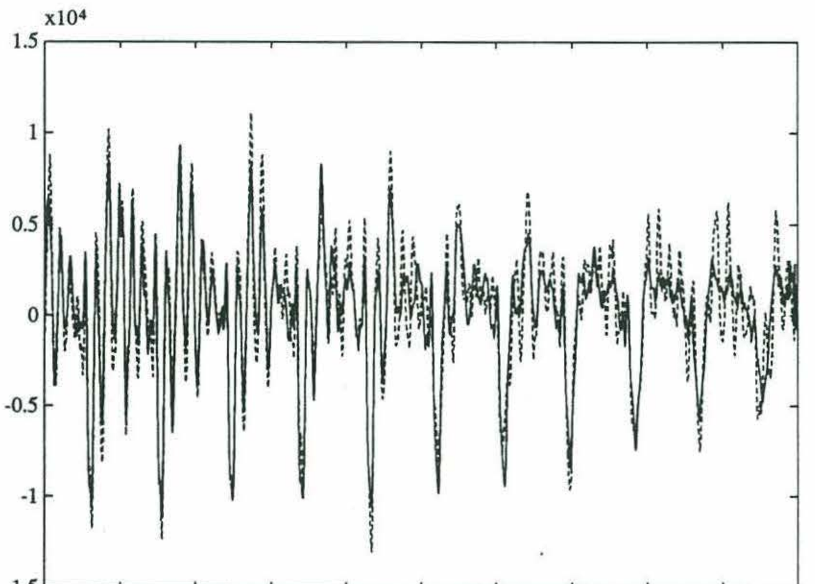

Figure 5-3: Actual(solid) and Estimated(dashed) Speech

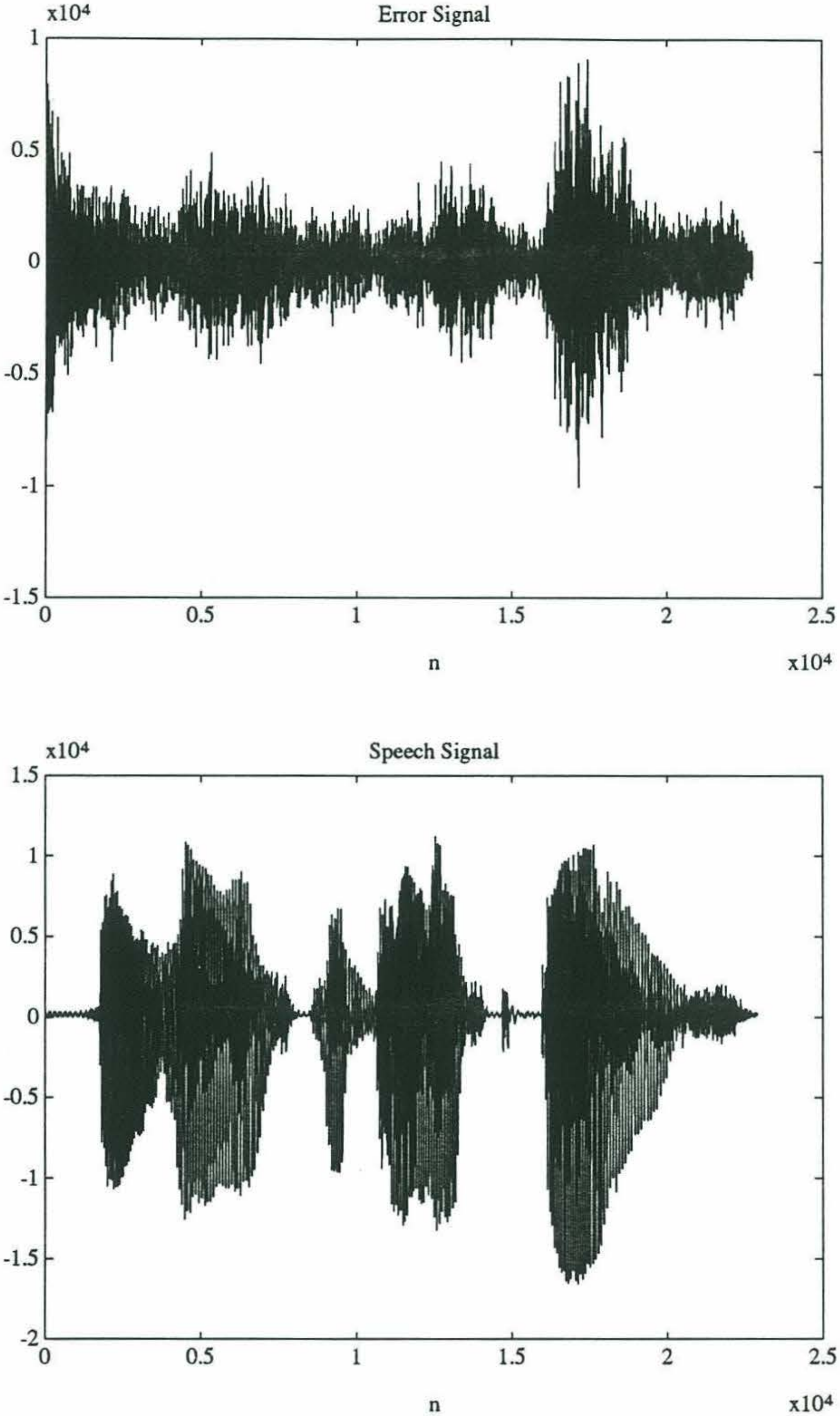

Figure 5-4: Error Signal and Actual Signal for Gradient-based Algorithm

using the following formula:

$$\mathrm{SER[n]} = 10\log_{10}\left(\frac{\sum_{k=n-99}^{n}(s[k])^2}{\sum_{k=n-99}^{n}(s[k]-\hat{s}[k])^2}\right). \tag{5.1}$$

Figure 5-5 also shows the desired signal to facilitate correlation of the performance against the signal being estimated. From this, it is clear that the places the algorithm did worst were during the silence periods, and transitions into and out of the silence periods. During the voiced segments, the algorithm usually performed better than 10 dB enhancement, and occasionally approached 20 dB of SER during some of the more stationary voiced segments. In qualitatively listening to the estimate, the improvement was quite noticeable when compared against the original sensor data. The background noise came up quite a bit between words, but the actual words came through clearly. However, the estimate does not sound as good as the estimate obtained using the exact solution M-step.

Looking closely again at Figure 5-1, and correlating against the desired signal, it is clear that the estimates wavered noticeably during several of the transition periods of $s[n]$, most noticeably around $n = 17,000 - 18,000$. This suggests that perhaps the estimate of $\boldsymbol{a}$ made by the algorithm was more sensitive to the nonstationarities than the estimate of g_s. Originally, the step sizes were chosen such that the estimates would converge reasonably quickly on the correct values, but still remain reasonably stable once getting there. The behavior of the estimate of $\boldsymbol{a}$ indicates perhaps $\tilde{\delta}_s$, the step size for g_s, should be made larger relative to δ_a, the step size for $\boldsymbol{a}$. If it is assumed in advance that the filter is time-invariant, but the signal is time-varying, the step sizes should be set up so the estimates of the signal parameters are more likely to adapt to non-stationarities than the estimate of the filter impulse response is.

In summary, the gradient-based algorithm gives significant enhancement at much lower computational cost, though it does not equal the performance of the precise solution M-step. However, the gradient-based algorithm adapts much better when starting from no prior information about the filter coefficients.

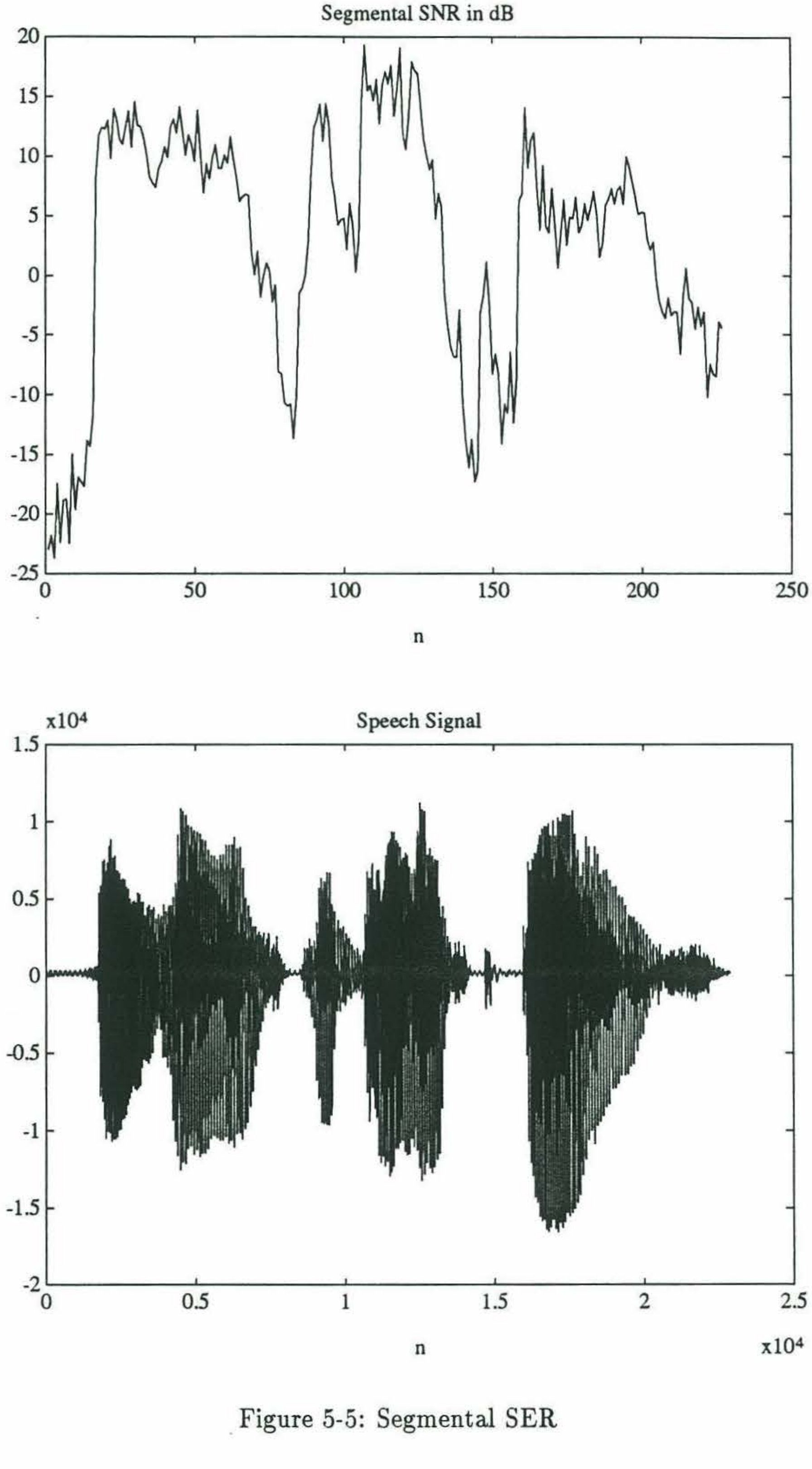

Figure 5-5: Segmental SER

Chapter 6

Conclusions and Future Directions

This chapter recaps the results of the thesis and suggests directions or topics for further investigation based on the work done here.

6.1 Conclusions

The most important result was the empirical verification that the algorithm at least works for favorable data sets. As no theoretical proofs of convergence existed for the sequential, causal form of the algorithm, this assurance that it actually performed signal enhancement was crucial.

Next, the desired signal was changed from a stationary AR signal to speech. Also the simplistic coupling filters were replaced with relatively long simulated room impulse responses. The algorithm still worked under well under these conditions, though the performance was found to be about 5 dB below the performance of the noncausal, iterative, block algorithm described in [TR 532] on the same data.

Finally, the gradient-based algorithm was tested on the speech with the same simulated room acoustics. It was found that while it was roughly fifty times cheaper, computationally speaking, the gradient-based M-step did not perform as well as the algorithm using the precise solution M-step. It did fairly well during the actual voiced sections of the signal, but the perfomance fell off badly during the silent sections. It was also discovered that the gradient-based algorithm adapted much better from initial conditions assuming no prior information about the filter coefficients.

6.2 Future Directions

One perplexing unanswered question is why the algorithm cannot estimate both filters simultaneously, but needs to have one filter known and held fixed. Very preliminary inconclusive results suggest the gradient-based algorithm may solve this. It is possible one filter may need to "lock on" before the other converges, and it may just take a long time for that first filter to converge. Perhaps letting the algorithm run longer than any of the experiments described in this thesis would allow that to happen.

Another issue to be examined is the SNR threshold at which this enhancement algorithm stops working. In general, enhancement algorithms often perform reasonably well until the SNR measured at the sensors goes below some threshold, at which point they fail catastrophically. A determination of this threshold and comparison against the boundary for other algorithms would be interesting.

The issue of more sophisticated strategies for choosing and modifying the step sizes for the gradient-based algorithm is also an open question. As seen in Chapter 5, the signal parameters should be more sensitive to nonstationarities than the filter coefficients. Alternatively, a crude silence detection algorithm based on power could be used to determine step size, and greatly decrease all the step sizes during silent sections.

Preliminary experiments using colored noise as the corrupting noise, $w[n]$, have indicated this algorithm may not be useful for enhancement when the noise fails to meet the whiteness assumption. Specifically, the noise used was aircraft noise from F15 and King Air jets, recorded in the cockpit. Both of these are fairly narrow-band noises. In general, narrow-band noise can lend itself well to cancellation or enhancement algorithms, but appears to cause the one discussed in this thesis considerable difficulty. However, modifications of the algorithm for this problem are conceivable, and could perhaps give practically useful enhancement algorithms for colored corrupting noises.

Appendix A

Matlab Program Listing

```
% remtest.m
% John Buck
% 6/27/90

% script to run the time-adaptive em algorithm using efficient
% Kalman smoothing and recursively-implemented m step.  this
% version of the script is designed for remote runs overnight
% from whoi, so it should be fed into matlab on the command
% line after changing the filenames in the initial string
% variables for the input and output.

% filename variables
% infile should be the pathname of the input data
% outfile should be the pathname of the file where matlab
%  will save all relevant variables (ie parameters and signal
%  estimates) when it finishes
infile = '/fs/TMP/rosencra/z128.mat'
outfile = '/fs/TMP/rosencra/oct11e1.mat'

global z1;
global z2;

% fetch microphone signals from saved variables
rem_get_mikes(infile);

% initialize the myriad parameters
[a,q,b,r,alpha,p,x,xxt,P,g1,g2,gs,gw,gamma1,gamma2,gammas,...
gammaw,t_stop]= rem_init;

% the following section is needed for the precise solution
% m-step, but not for the gradient algorithm

% initialize state variables for recursive algorithm
%A = zeros(p,p);
%B = zeros(p,1);
%C = zeros(q+1,q+1);
%D = zeros(q+1,1);
%E = zeros(r+1,r+1);
%F = zeros(r+1,1);
```

```
%G = 0;
%H = 0;
%J = 0;

% find start time
t_start = max(r,q)+2;

% start monster iteration
for t=t_start:t_stop

  z = [z1(t);z2(t)];


% signal estimation step
  [x,P] = effestep(x,P,z,p,r,q,alpha,a,b,gs,gw,g1,g2);

% gradient based estimation
[alpha,gs,gw,a,g1,b,g2] = gradest(x,P,alpha,0.0001,gs,0.0001,...
gw,0.0001,a,0.001,g1,z,0.0001,b,0.0001,g2,0.0001,p,q,r);


% the following is the precise solution m-step.
%  [A,B,alpha,C,D,a,E,F,b,gw,G,gs,H,g1,J,g2] = ...
%     recmax(A,B,gammas,r,p,C,D,x,P,gamma1,q,z,E,F,gamma2,gw,gammaw,...
%       t-t_start+1,G,H,J,alpha,a,b,g1);

  % save all current values to we can look at time profiles of things.
  % this version assumes  g1 and g2 are not running free, but fixed
  % at their true values.

  % save one in ten values of a, alpha, gs
if (rem(t-t_start,100) == 0)
gs_t(((t-t_start)/10)+1) = gs;
a_t(:,((t-t_start)/10)+1) = a;
alpha_t(:,((t-t_start)/10)+1) = alpha;
end


  % save current sample for speech. ie s(t)
  speech(t-t_start+1) = x(r+1,1);

% repeat ad nauseum
end

g1_t = 0;
g2_t = 0;
gw_t = 0;
b_t = 0;

save_results(outfile,speech,a,b,alpha,gs,gw,a_t,alpha_t,gs_t,gw_t,g1_t,g2_t,b_t);

'done'
```

```
function [x,P] = effestep(x,P,z,p,r,q,alpha,a,b,gs,gw,g1,g2)
% EFFESTEP  Jigsaw puzzle style efficient e step
% assumes r>p

% this implements the kalman filter taking advantage of the
% sparse state-space propagation matrix

% Propagation
x(1:r) = x(2:r+1);
x(r+2:r+q+2) = [x(r+3:r+q+2);0];
x(r+1) = -alpha'*x(r-p+1:r);
P(1:r,1:r) = P(2:r+1,2:r+1);
P(1:r,r+2:r+q+1) = P(2:r+1,r+3:r+q+2);
P(r+2:r+q+1,1:r) = P(1:r,r+2:r+q+1)';
P(r+2:r+q+1,r+2:r+q+1) = P(r+3:r+q+2,r+3:r+q+2);
P(1:r+q+1,r+q+2) = zeros(r+q+1,1);
P(r+q+2,1:r+q+1) = zeros(1,r+q+1);
P(r+q+2,r+q+2) = gw;
P(1:r,r+1) = -P(1:r,r-p+1:r)*alpha;
P(r+1,1:r) = P(1:r,r+1)';
P(r+1,r+1) = alpha'*P(r-p+1:r,r-p+1:r)*alpha+gs;
P(r+1,r+2:r+q+1) = -alpha'*P(r-p+1:r,r+2:r+q+1);
P(r+2:r+q+1,r+1) = P(r+1,r+2:r+q+1)';

% Up-dating Equations

% make gains first

w11 = a(1:q)'*P(r+2:r+q+1,r+2:r+q+1)*a(1:q)...
 -2*a(1:q)'*P(r-p+1:r,r+2:r+q+1)'*alpha...
 +alpha'*P(r-p+1:r,r-p+1:r)*alpha+...
 gs+a(q+1)*a(q+1)*gw + g1;

w22 = b(1:r)'*P(1:r,1:r)*b(1:r)...
 -2*b(r+1)*b(1:r)'*P(1:r,r-p+1:r)*alpha + ...
b(r+1)*b(r+1)*(alpha'*P(r-p+1:r,r-p+1:r)*alpha+gs) + gw + g2;

w12 = a(1:q)'*P(1:r,r+2:r+q+1)'*b(1:r)...
 -b(r+1)*a(1:q)'*P(r-p+1:r,r+2:r+q+1)'*alpha...
-b(1:r)'*P(1:r,r-p+1:r)*alpha+...
b(r+1)*(alpha'*P(r-p+1:r,r-p+1:r)*alpha+gs)+a(q+1)*gw;

W = [w11,w12;w12,w22];

Q(1:r,1:2) = [-P(1:r,r-p+1:r)*alpha+P(1:r,r+2:r+q+1)*a(1:q),...
 P(1:r,1:r)*b(1:r)-P(1:r,r-p+1:r)*alpha*b(r+1)];
Q(r+1,1:2) = [alpha'*P(r-p+1:r,r-p+1:r)*alpha+gs-...
alpha'*P(r-p+1:r,r+2:r+q+1)*a(1:q),-alpha'*P(1:r,r-p+1:r)'*b(1:r)+...
b(r+1)*(alpha'*P(r-p+1:r,r-p+1:r)*alpha+gs) ];
Q(r+2:r+q+1,1:2) = [-P(r-p+1:r,r+2:r+q+1)'*alpha+...
P(r+2:r+q+1,r+2:r+q+1)*a(1:q),...
P(1:r,r+2:r+q+1)'*b(1:r)-P(r-p+1:r,r+2:r+q+1)'*alpha*b(r+1)];
Q(r+q+2,1:2) = [a(q+1)*gw,gw];
```

```
K = Q*inv(W);

%new estimates
P = P - K*Q';
x = x+K*[z(1)+alpha'*x(r-p+1:r)-a(1:q)'*x(r+2:r+q+1);...
z(2)+b(r+1)*alpha'*x(r-p+1:r)-b(1:r)'*x(1:r)];
```

```
function [newalpha, newgs, newgw, newa, newg1, newb, newg2] = ...
gradest(x,P,alpha,delalpha,gs,dels,gw,delw,a,dela,g1,z,del1,...
b,delb,g2,del2,p,q,r)
% GRADEST  This function implements Udi's equations for the gradient based
%          parameter estimation in the two microphone case.
%     it uses intermediate variables such as sps, spspalpha
%     to minimize computation

% note: several lines are commented out as they were for quantities
% not estimated in the experiments in the thesis
sps = x(r-p+1:r)*x(r+1)+P(r-p+1:r,r+1);
spspalpha = (x(r-p+1:r)*x(r-p+1:r)'+P(r-p+1:r,r-p+1:r))*alpha;
newalpha = alpha - (delalpha/gs)*(sps+spspalpha);
newgs = (1-dels/2)*gs + (dels/2)*(x(r+1)^2+P(r+1,r+1)+...
2*alpha'*sps+alpha'*spspalpha);
%newgw = (1-delw/2)*gw+(delw/2)*(x(r+q+2)^2+P(r+q+2,r+q+2));
wqz1 = x(r+2:r+q+2)*z(1);
wqs = (x(r+2:r+q+2)*x(r+1)+P(r+2:r+q+2,r+1));
wqwqa = (x(r+2:r+q+2)*x(r+2:r+q+2)'+P(r+2:r+q+2,r+2:r+q+2))*a;
newa = a+(dela/g1)*(wqz1-wqs-wqwqa);
%newg1 = (1-del1/2)*g1 + (del1/2)*(z(1)^2 - 2*z(1)*x(r+1)+...
% x(r+1)^2+P(r+1,r+1)-2*a'*wqz1+2*a'*wqs+a'*wqwqa);
%srz2 = x(1:r+1)*z(2);
%srw = x(1:r+1)*x(r+q+2)+P(1:r+1,r+q+2);
%srsrb = (x(1:r+1)*x(1:r+1)'+P(1:r+1,1:r+1))*b;
%newb = b+(delb/g2)*(srz2-srw-srsrb);
%newg2 = (1-delb/2)*g2+(del2/2)*(z(2)^2-2*z(2)*x(r+q+2)+x(r+q+2)^2+...
% P(r+q+2,r+q+2)-2*b'*srz2+2*b'*srw+b'*srsrb);
newgw = gw;
newg1 = g1;
newg2 = g2;
newb = b;
```

```
function [newA,newB,newalpha,newC,newD,newa,newE,newF,newb,newgw,...
newG,newgs,newH,newg1,newJ,newg2]...
  = recmax(A,B,gammas,r,p,C,D,x,P,gamma1,q,z,E,F,gamma2,gw,gammaw,t,G,...
H,J,alpha,a,b,g1)
% RECMAX    does max-likelihood estimation using recursively
%           updated state variables

% this file contains the equations for the precise solution m-step.
% several lines involving quantities not estimated are included.

newA = gammas*A + x(r-p+1:r)*x(r-p+1:r)'+P(r-p+1:r,r-p+1:r);
newB = gammas*B + x(r-p+1:r)*x(r+1)+P(r-p+1:r,r+1);
newalpha = - inv(newA) * newB;


newC = gamma1*C + x(r+2:r+q+2)*x(r+2:r+q+2)'+P(r+2:r+q+2,r+2:r+q+2);
newD = gamma1*D+z(1)*x(r+2:r+q+2)-x(r+2:r+q+2)*x(r+1)-P(r+2:r+q+2,r+1);
newa = inv(newC) * newD;

%newE = gamma2*E + xxt(1:r+1,1:r+1);
%newF = gamma2*F + z(2)*x(1:r+1)-x(1:r+1)*x(r+q+2)+P(1:r+1,r+q+2);
%newb = inv(newE) * newF;
%newb = newb(10:-1:1)';
newb = b;

%if (gammaw==1)
% newgw = (gw*(t-1) + x(r+q+2)*x(r+q+2)+P(r+q+2,r+q+2))/t;
%else
% gammawtot = gammaw^t;
% newgw = (gw*(gammaw-gammawtot)+(1-gammaw)*...
%  x(r+q+2)*x(r+q+2)+P(r+q+2,r+q+2))/(1-gammawtot);
%end
newgw = gw;

newG = gammas*G+x(r+1)*x(r+1)+P(r+1,r+1);
if (gammas==1)
newgs = (newG+newalpha'*newB)/t;
else
newgs = (1-gammas)*(newG+newalpha'*newB)/(1-gammas^t);
end

%decay g1 and g2 assuming estimates will grow more accurate
%if (t<=1000)
% newg1 = 2002000-t*2000;
%else
% newg1 = g1;
%end
newg1 = g1;
newg2 = newg1;


%newH = gamma1*H+z(1)*z(1)-2*x(r+1)*z(1)+x(r+1)*x(r+1)+P(r+1,r+1);
%newg1 = newH-newa'*newD;
%if (gamma1==1)
```

```
% newg1 = newg1/t;
%else
% newg1 = (1-gamma1)*newg1/(1-gamma1^t);
%end
%
%newJ = gamma2*J+z(2)*z(2)-2*x(r+q+2)*z(2)+x(r+q+2)*x(r+q+2)+P(r+q+2,r+q+2);
%newg2 = newJ - newb'*newF;
%if (gamma2==1)
% newg2 = newg2/t;
%else
% newg2 = (1-gamma2)*newg2/(1-gamma2^t);
%end
```

```
function rem_get_mikes(mikefile)

% REM_GET_MIKES reads the inputs for from mikefile for the em algorithm

mikefilel = length(mikefile);
loadcom = 'load ';
loadcom(6:6+mikefilel-1) = mikefile;
eval(loadcom);
```

```
function [a,q,b,r,alpha,p,x,xxt,P,g1,g2,gs,gw,delta1,delta2,gammas,...
gammaw,t_stop]= rem_init
% INIT_PARAMS Initializes the monster list of parameters for
% the E-M algorithms, where z1 and z2 are the
% signals from the two microphones.


% Udi suggests using the first r+1 values from mike 1 as the initial
% guess at the speech, and the first q+1 from mike 2 as the initial
% guess for the noise.

% use widrow-like lms to get initial a and alpha

N = 300;
q = 127;
for (i=1:q+1)
     yy2(:,i) = [zeros(1,q+1-i),z2(1:N+i-q-1)]';
end
a = yy2\(z1(1:N)')
clear yy2
p = 2;
for (i=1:N+p-1)
     if (i<(q+1))
        shat(i) = z1(i)-a'*([zeros(1,q+1-i),z2(1:i)]');
     else
        shat(i) = z1(i)-a'*(z2(i-q:i)');
     end
end
size(shat)
for i = 1:p
  ss2(:,i) = shat(p+1-i:N-i)';
end
alpha = ss2\(shat(p+1:N)');
clear ss2
clear shat
%need to reverse and negate alphas to match udi
alpha = -alpha(p:-1:1)

q = length(a)-1;
a = a(q+1:-1:1)';
load b128
r = length(b)-1;
b = b(r+1:-1:1)'

g1 = 100;
g2 = 100;
gs = 1e6;
gw = 1.667e8;
P=[gs*eye(r+1),zeros(r+1,q+1);zeros(q+1,r+1),gw*eye(q+1)];
x = [z1(1:r+1), z2(1:q+1)]';
xxt = x*x';
xxt = xxt+P;
delta = 0.995;
delta1 = delta;
```

```
delta2 = delta;
gammas = 0.995;
gammaw = 0.995;
t_stop = length(z1);
```

```
function save_results(outfile,speech,a,b,alpha,gs,gw,a_t,alpha_t,gs_t,gw_t,...
      g1_t,g2_t,b_t)

outfilel = length(outfile);
savecom = ['save ',outfile,' speech a b alpha gs gw a_t alpha_t gs_t ...
        g1_t g2_t gw_t b_t'];
eval(savecom)
```

Bibliography

[Anderson 79] B. O. Anderson and J. B. Moore, *Optimal Filtering.* Prentice-Hall, Inc. Englewood Cliffs, N.J., 1979.

[Dempster 77] A.P. Dempster, N.M. Laird and D.B. Rubin. "Maximum Likelihood From Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society.* Series 3g, pp. 1-38. 1977.

[Elvis 91] Elvis Presley. Personal Communication. Cambridge, Mass. May, 1991.

[Feder 89] Meir Feder, Alan V. Oppenheim and Ehud Weinstein. "Maximum Likelihood Noise Cancellation Using the EM Algorithm." *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-37, pp. 204-216, 1989.

[Gelb 74] Arthur Gelb, ed. *Applied Optimal Estimation.* M.I.T. Press. Cambridge, Mass. 1974.

[RS 78] Lawrence Rabiner and Ronald Schafer. *Digital Processing of Speech Signals.* Prentice-Hall, Inc. Englewood Cliffs, N.J. 1978.

[TR 532] Meir Feder. *Statistical Signal Processing using a class of Iterative Estimation Algorithms.* MIT Research Lab for Electronics Technical Report 532. Cambridge, Mass. September, 1987.

[TR 560] E. Weinstein, A. V. Oppenheim and M. Feder. *Signal Enhancement Using Single and Multi-Sensor Measurements.* MIT Research Lab for Electronics Technical Report 560. Cambridge, Mass. November, 1990.

[Widrow 75] Bernard Widrow *et al.,* "Adaptive Noise Cancelling: Principles and Application." *Proc. IEEE,* vol. 63, pp. 1692-1716, 1975.

[Widrow 85] Bernard Widrow and Samuel Stearns. *Adaptive Signal Processing.* Prentice-Hall, Inc. Englewood Cliffs, N.J. 1985.